Received: May 3, 2024. Revised: June 1, 2024.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Time-Aware Attention and Knowledge Graph Embedding in Deep Learning Model for Improving Customer Preference based Recommendations

Natarajan Sathya[1]* Subramaniam Duraisamy[1] Thangamuthu Latha Maheshwari[2]

[1]*Department of Computer Science, Chikkanna Government Arts College, Tiruppur - 641602, Tamil Nadu, India*
[2]*Department of Computer Science and Engineering, Sri Krishna College of Engineering and Technology, Coimbatore- 641008, Tamil Nadu, India*
* Corresponding author's Email: sathya29natarajanphd@gmail.com

Abstract: Recognizing consumer interests and satisfaction is crucial for organizations to thrive in a competitive market. A strong marketing strategy aids companies in reaching customers and building a reputation for their products. Presently, numerous internet platforms employ recommender systems on a commercial basis. Customer preferences are generally obtained behaviours of users from online shopping which is used to recommend the products. Deep Learning (DL) models have been used for efficient customer preference recommendation, but these models don't consider secondary information such as Knowledge Graph (KG) and comment information. Hence, in this paper, Time Aware Attention and KG embedded Deep Network for Recommendation (TAK-DepNR) is proposed by combining Convolutional Neural Network (CNN) and Graph Convolutional Network (GCN). In this method, the user similarity, item similarity and latent period are modelled using Matrix Factorization (MF) model. The MF generates long and short preferences which are then fed into the CNN and GCN models to produce dynamic user preferences. The CNN extracts higher-level characteristics among the user and item from MF, then reduces to lower-order features. The GCN aids to train the interpretation of each node and encodes the KG based Global Convolutional Feature Map (KGGCFM) for CNN through the shared features. This creates a major link between the products and allowing for more extensive item feature description for user preference graph. The CNN will be integrated with GCN using its squeezed units to learn the high-level feature relations between recommendation systems and KG entities. Finally, dynamic user preferences from CNN and GCN are passed into Fully Connected (FC) layer constructed network to recommend the items. The experimental results show that the TAK-DepNR realizes average precision of 89% and 88% compared to the other models like Immersive Graph Neural Network (IGNN), Attentional Factorization Machine with Review-Based User-Item Interaction (AFMRUI), Item Collaborative Filtering with RNN (ICFRNN), Deep Reinforcement Recommender System with Maximum pooling layers (DRR-Max) and Sentiment Analysis with Bidirectional Long Short Term Memory (SA-BiLSTM) on Amazon Digital Music Dataset and Book Crossing Dataset.

Keywords: Customer preferences, Matrix factorization, Convolutional neural network, Graph convolutional network, Recommendation system.

1. Introduction

In recent days, the abundance of products and commodities in the market presents challenges for businesses and consumers. Consumers have numerous alternatives, but choosing the best one is challenging [1]. The free market structure makes it difficult to attract new customers and maintain existing ones, leading to fierce competition and difficulty in gaining profits in share markets [2]. Therefore, it is critical for businesses to adapt to client preferences and provide items that satisfy their requirements.

Recommendation systems provide personalized recommendations based on consumer interests and preferences, addressing challenges in real-time applications such as online social networks, digital government, online shopping and online education, among others [3]. CF is a powerful method for developing recommendation systems [4] with two main categories: Model-based CFs and Memory-based CFs. Model-based methods offer numerical recommendations, while memory-based methods like customer and product-based CFs delivers an approximate unidentified opinions by compiling similar client or product preferences [5].

Several instantaneous big-data concepts and the exponential growth of data sets have recently presented formidable obstacles to the conventional approaches of recommender systems [6]. It has become critical for obtaining massive amounts of data due to the need to evaluate the customer's demands and preferences. Most suggestion engines may work better with fewer details, but they struggle to fit into the big data perspective due to the high cost of acquiring massive volumes of data [7].

Artificial Intelligence models like Machine Learning (ML) and Deep Learning (DL) are utilized to analyze consumer preferences and develop effective product recommendation systems [8]. ML utilizes the numerical algorithms to identify data correlations and automate hypothesis development [9, 10]. However, ML faces challenges in extracting useful data from large databases and requires distinct feature selection before training, resulting in computational complexity.

Deep Learning (DL) models are introduced to analyze the customer preferences and develop efficient recommendation systems for specific products [11]. DL models learn the beneficial characteristics from customer reviews aiding in making helpful product recommendations to consumers who manage large amounts of data. DL is more effective than feature engineering due to its ability to dynamically identify complex characteristics and nonlinear depiction from raw data [12]. Various Deep Learning models, including CNN, Recurrent Neural Networks (RNN), Long-Short Time Memory (LSTM), Deep Belief Networks (DBN) and so on, can be utilized to address customer preferences challenges, thereby enhancing marketing technology and increasing company income by improving the marketing methods for various goods and services [13].

From this perspective, various research works have been developed for the efficient recommendation system based on customer preferences. A two-stage recommendation model using Attention-based Deep Neural Ranking (A-DNR) was developed [14], considering historical, current and contact time for customer's preferences. A novel Neural Network (NN) was designed to suit linear and nonlinear interaction patterns. A ranking list was determined based on these scores. But, the inclusion of auxiliary information such as KGs were not considered in this model which lowers the recall rate. A Particularized Suggestion System (PSS) was created using the virtual marketing environment and IGNN [15] which enhances marketing environments and customer needs for convenient purchasing exposures and product features. But, it has high computational complexity and provides lower precision values.

To address the above issues, TAK-DepNR is developed to learn the KG information with lesser complexities by integrating CNN and GCN model. In this method, the user similarity, item similarity and latent period are modelled using MF model. The long and short preferences derived from MF are individually fed into CNN and GCN model to form dynamic user preferences. The CNN captures higher-order features between user and item from MF, then reduces to lower-order features. Two changes are made to the CNN network from the VGG16 model: first, a global average pooling layer is added to minimize spatial dimensions and second, the max-pooling layer will be removed from the final convolution blocks. Using the shared features, the GCN encodes the KGGCFM of the CNN and aids in learning the interpretation of each node. In order to convey item attributes in the recommendation in more depth, this establishes a substantial relationship between the items. Using its squeezed units, the CNN will integrate with GCN. CNN is mainly employed to gather high-dimensional attributes and train the high-level feature associations between suggested items and entities in the KG dynamically. Finally, dynamic features from CNN and GCN are passed into FC layer to identify the missing KG data and reduces the complexity of the constructed network for an efficient recommendation system.

The following remaining portions are planned: Section II discusses previous research on user-item recommendation systems. Section III discusses the TAK-DepNR approach, while Section IV demonstrates its evaluation performance. Section V summarizes the whole article and offers suggestions for further development.

2. Related works

A diversed behavior recommendation approach using used Graph Transformer19 CF (GTCF) was introduced [16] for user-item recommendation system. But, this model results in lower precision and recall value due to the overfitting issue.

A product suggestion engine called SmartTips was constructed [17] which leverages attribute sentiment analytics, gauge user sentiments and search for current preferences using review text and votes for recommendation. But, only subjective layout was anticipated in suggesting the items opinions which lowers the average precision rate.

A Composite-Bridged E-Commerce Knowledge Suggestion Engine (CBEKSE) was created [18] based on the DL model to overcome geographical difficulties and increase suggestion efficacy. However, this model was easy prone to certainty and numerical latent issues significantly lowers the recall efficiency.

A Deep Neural Network (DNN) model was presented [19] to analyze consumer behavior and predict commodity recommendations. The model uses Genetic Algorithm to extract customer purchase behavior rules and DNN to classify transaction data samples, enabling product recommendations on consumer behavior. However, this model was trained with limited dataset limiting the precision rate.

A neural MF model called EINMF was developed [20] to predict user interest using both explicit and implicit input data for better recommendation. ML models accurately indicate user interests through individual suggestions and learning information from user-item collaboration. However, lower Mean Average Precision (MAP) was resulted due to the large residual variable embedding size.

A AFMRUI model was presented [21] for recommendation. This approach improves rating prediction and recommendation promotion by learning the connections between user-item features using a combination of attention networks and bidirectional gated recurrent units. But this model results with lower precision rate when iterated on various interest nodes.

A deep ensemble learning method called SARWAS was presented [22] for effective product-customer supportive systems. This system employs neural network with hidden layers and activation functions to identify users and items based on reviews and ratings, forecasting combined scores for unknown combinations. However, this model produces lower precision results on larger dataset.

An ICFRNN model was devised [23] which transforms the textual data into the numeric value for measuring item–item similarities for the efficient recommendation system. The model distributes text to relevant classes, computes item rating scores from reviews to define item uniformity using the scores. However, recall was degraded when the latent factor embedding dimension was too large.

A DRR-Max was developed [24] for real-time recommendation system based on the user preferences. This system utilizes a state iteration module to collect customer's historical dat. Actor-importance approach was used for dynamic recommendations through both offline and online authentic instruction. But, this approach proved inadequate precision values in examining consumer preferences across a more extensive dataset.

A new user recommendation model was SA-BiLSTM was developed [25] for efficient recommendation system. In this method, sentiment model was built using Bi-LSTM, GloV vectorization and collaborative filtering techniques for more precise and personalized recommendations. But, MAP values were lowered exceeded over the fluctuation range in predicting the short and long term demands.

Table 1. List of Notations

Notations	**Description**
U	Group of Users
I	Set of Item
A	Number of users
B	Number of items
D	Implicit feedback
g	Knowledge Graph
W	Weight Parameter
V_m	Product Value with m dimension
V_Q	Initial product feature representation
Y_u^S	Short-Term Preferences of User u
$T.$	Time Interval
Y_u^L	Long-Term Preferences of User u
$H(u)$	Historical Interaction Items of User u
L	Layer Depth
σ	Nonlinear activation function
$w^{(l)}$	Model Weight
$B^{(l)}$	Bias Layer
X	Input Feature Map
e ; h	Width, Height
c	Number of Channels
U_N	User Latent Vector
F_N	Feature Latent Vector
P	Recommended Score
h_i^l	Feature Node Representation in l^{th}layer
$G_k()$	Message Aggregation Function
$\mathcal{N}_i^r$	Neighbourhood nodes of i with relation r
$\mathcal{P}$	Trainable Parameter
r^l	Feature vector Relationship with l layer
$\tilde{t}$	Tail Entity
$\tilde{h}$	Head Entity
V	Commodity Interaction
E	Corresponding Entity
F_M	Feature Matrix
G_i	CNN branch features
h_i	GCN branch features
$\oplus$	Element-wise addition
δ	Fusion ratio parameter
$\mathcal{L}_{KGE}$	Knowledge Graph Embedding Loss
$\mathcal{L}_{DREC}$	Depth Recommendation Loss
$\mathcal{E}$	Number of edges

Received: May 3, 2024. Revised: June 1, 2024.

3. Proposed methodology

The development of the proposed TAK-DepNR model for the efficient recommendation system is detailed in this section. Table 1 provides the notation list.

Fig. 1 shows the high-level structure of this work. This model is composed of different sections, (1) MF model to generate the long and short term preferences, (2) CNN structure to learn high order features, (3) GCN model to learn each node representation and KG relationship, (4) CNN and its compressed units for sharing the features (5) dynamic features to FC layer.

3.1 Problem definition

The recommendation algorithms employ the following notation: i.e., user batch $U = u_1, u_2, .., u_A$ and collection of item $I = i_1, i_2, .., i_B$, where A and B represent the quantity of users and items respectively. The corresponding producers are referred to as products, films, or TVs. The user's latent response on the product is denoted by D and the relationship matrix among the product and the customer is as $D = \mathcal{R}^{A*B}$. The user's action i.e, liking, viewing or clicking are all indicated when $d_{ui} = 1$. In case of $v_{ui} = 0$, it means the user didn't response to the respective objects. To improve the suggestions from the customers, the KG is derived in this paper. In the KG, $g = (head, relation, tail)$ constitutes to a triples which denote the relationship among two parts. A more accurate portrayal of each items in assist to KG will strengthen the overall recommendations. Thus, this study aimed to promote an item $i \in I$ to $u \in U$ using item knowledge in given a collection of users U and items I fused with KG.

3.2 Modelling long and short term preferences using MF

Using user similarity, item similarity and time latency, this model successfully derived long and short-term desires using MF. The MF model plays a significant role in determining the flexible user interests by extracting both high and lower-order attributes, which are defined as the bilateral combinations of high-order variables. The extraction of the commodities characteristics Cf_{MF} using the MF paradigm is mathematically expressed as

$$Cf_{MF} = W_0 + \sum_{m=1}^{n} W_m V_m + \sum_{m=1}^{n-1} \sum_{Q=m+1}^{n} W_{mn} V_m V_Q \quad (1)$$

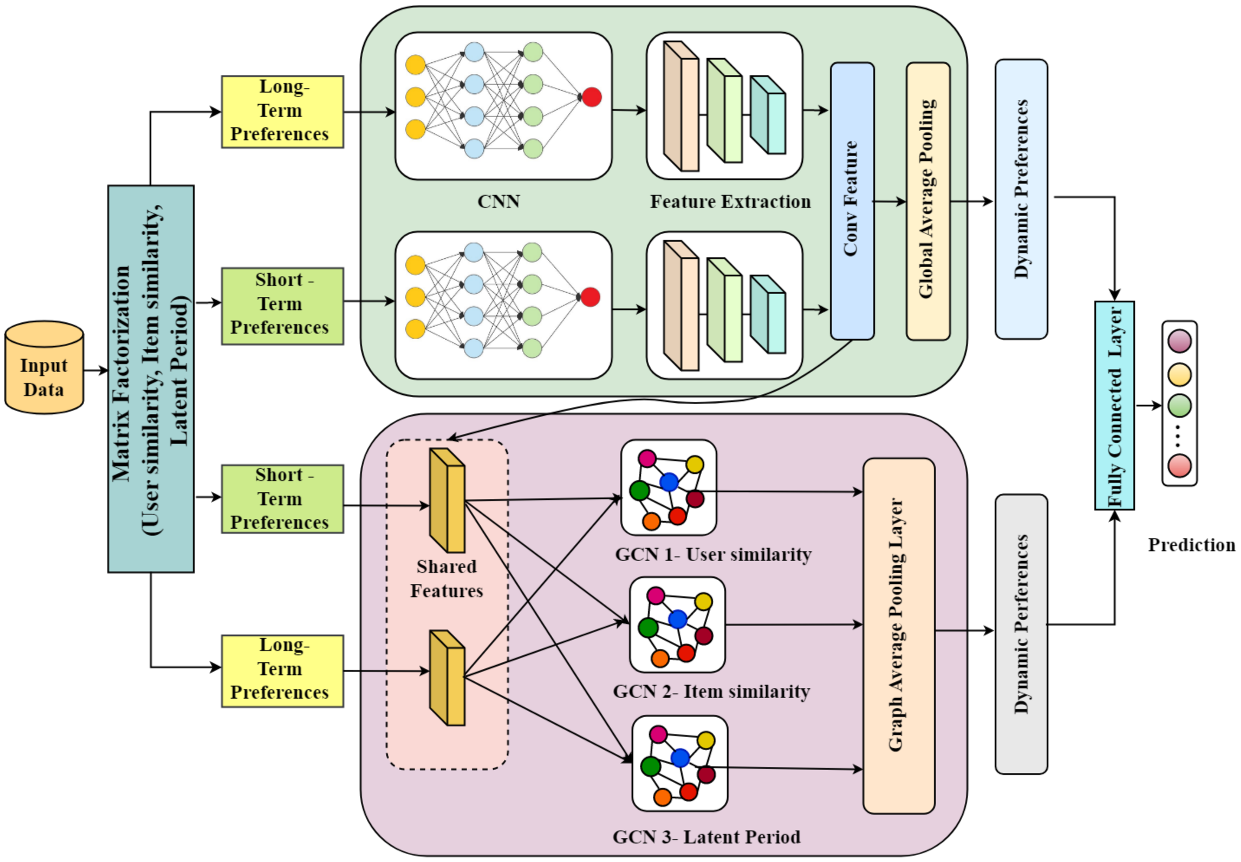

Figure. 1 Overall Flow of research work

The model has to acquire the variable W which is represented by V_m in the previous Eq. (1), to indicate the m^{th} dimension of the goods and V_Q is the first feature vector depiction of the item. These are the user's immediate interests based on the MF highly emphasize on goods from their previous history of interactions (i) as given in Eq. (2),

$$Y_u^S = \sum_{i=1}^{m} MF\ (i)\ T_i \tag{2}$$

In above Eq. (2), Y_u^S defines the user u in short-term interests and $MF(i)$ is the MF weight of the most current co-relative item i in relevant to time T. The following is the model for the user's extended preferences using the user's integrated vector in prospective space in Eq. (3):

$$Y_u^L = \frac{\sum_{i\in l(u)} T_i}{|(H(u))|} \tag{3}$$

The long-term desires of user u are represented by Y_u^L, while all of user's previous interactive items denoted by $H(u)$.

3.3 CNN model for higher order feature extraction

Using MF as a source, the CNN in this model extracts higher-level information about the customers and the products, which it then reduces to lower-order features. The CNN will be integrated with GCN using its squeezed units in which CNN will automatically acquire the high-level relationships among products and entities in the KG. The constructed CNN model is adjusted to be a concised and efficient VGG16. A max-pooling procedure concludes each of the five convolution blocks that include the convolutional layers.

The variables $h \times e \times c$, in which h and e stand for spatial coordinates and the c for channel numbers respectively which determines the magnitude of the feature map generated. Typically, a basic VGG16 network will conduct a feature map size of $1/64$ of the input (long and short term interests). It is observed that the network efficiently maintains precise knowledge within a significant spatial scale by making the feature map size $1/8$ that of the input.

This is because the CNN model must generate a multidimensional feature depiction of an item, which is divided using the MF as input. Feature representation $i_{CNN}^{(l)}$ of the l^{th} level item in high dimensions is defined in Eq. (4) as follows:

$$i_{CNN}^{(l)} = \sigma\left(w^{(l)} i_{CNN}^{(l-1)} + B^{(l)}\right) \tag{4}$$

Where σ is a nonlinear activation operation like Rectified Linear Units (ReLU) and L is the depth of the layer. The algorithm's weight and bias of the l^{th} layer which needs to be acquired are denoted as $w^{(l)}$ and $B^{(l)}$, correspondingly. Then, the process generates a dense feature vector with actual values which integrates the terminal three FC layers with an additional global mean pooling global mean pooling layer and calculates the mean of the height and breadth of the feature maps. The equations of global mean pooling are mentioned in Eq. (5) as follows:

$$G_{c,i} = \frac{\sum_{e=1}^{E}\sum_{h=1}^{H} x_{e,h,c,i}}{E*H} \tag{5}$$

The input feature map denoted by X, has breadth, height and channel numbers are represented by e, h and c. To the greatest extent feasible, the global mean pooling layer retains the fine-grained data in this suggestion system. Consequently, given the implicit feature vector U_N of the individual u and the residual feature F_N of the item i, the terminal formulation of the product $F_N = Cf_{MF} + i_{CNN}^{(l)}$ yields the following Eq. (6) for the suggested rating:

$$S(y) = sigmoid\ (U_N^T * F_N) \tag{6}$$

For combinations of high-order features, the CNN component employ the implicit feature vector U_N^T. To get suggestions for products, multiply the customer-product ranking matrix by the inversion of the item embedding matrix and the minimal order product score should be equal to the embedding vector product of two characteristics

3.4 Integrating GCN with KG

In order to preserve the coherence among relationships and entities, the KG embedding representation attempts to turn them into an uninterrupted vector space. In order to incorporate the KG, GCN is used in this article. A GCN may be defined as a framework for message distribution which is mathematically defined in Eq. (7) as follows:

$$h_i^{l+1} = \sigma\left(\sum_{Q\in A_i} G_k\left(h_i^l, h_Q^l\right)\right) \tag{7}$$

The message accumulative function is G_k () and h_i^l is the feature corresponding to the node V_m in the l^{th} layer. In particular, the typical GCN model $G_k\left(h_i^l, h_Q^l\right) = Wh_Q$ includes both node (entity) characteristics and edge (relation) features for KG embedding. The conversion of the GCN node feature encoding is expressed as shown in Eq. (8) as follows,

$$h_i^{l+1} = \sigma\left(\sum_{r\in R}\sum_{Q\in\mathcal{N}_i^r}\frac{1}{\mathcal{N}_i^r}\frac{1}{p_{i.r}}\left(W_r^l h_Q^l + W_0^l h_Q^l\right)\right) \quad (8)$$

The collection of adjacent nodes i under relation r is represented by $\mathcal{N}_i^r$ and the adaptable variable is $p_{i.r}$. The data and connections (edges) from one layer of nodes are used to generate the characteristics of the subsequent layer.The edge's feature encoding is illustrated in and Eq. (9) as follows,

$$r^{l+1} = \mathcal{P}\, r^l \quad (9)$$

Where $\mathcal{P}$ denotes the adaptable variable and r^l denotes the feature vector expression of the association on the l layer's relation. The FC layer is employed to detect the tail entity depiction by merging the association attributes in Eq. (10) is between the nodes and edges:

$$\tilde{t} = \mathcal{P}\begin{bmatrix}\tilde{t}\\ \tilde{r}\end{bmatrix} \quad (10)$$

Where $\tilde{t}$ is the vector illustration of the projected tail entity for the top entity $\tilde{h}$ and the relation $\tilde{r}$.

3.5 Cross and compress units

The convoluted ($conv$) features from the CNN model are fed as the partitioned features and given into the GCN model to understand the relationship characteristics between commodities and their related entities in the KG. The association feature matrix F_M between every product V and its matching entity E is mentioned in and Eq. (11) as follows:

$$F_M = V_{E^T} = \begin{bmatrix} V_0E_0 & \cdots & V_0E_0\\ \cdots & \cdots & \cdots\\ V_{n-1}E_0 & \cdots & V_{n-1}E_{n-1}\end{bmatrix} \quad (11)$$

Where, $V \in \mathcal{R}^{n*1}$, $E \in \mathcal{R}^{n*1}$, $F_b \in \mathcal{R}^{n*n}$. A cross-function is defined as the representation and modeling of each feature interaction V_iE_Q between the product and its related entity in the interaction feature matrix F_M. Eq. (12) and Eq. (13) illustrates the specified features of the product and entity after the interactions i.e., $\bar{V}$ and $\bar{E}$,

$$\bar{V} = CNN\left(F_M(p1) + {F_M}^T(p2) + b_V\right) \quad (12)$$

$$\bar{E} = CNN\left(F_M(p3) + {F_M}^T(p4) + b_E\right) \quad (13)$$

Where, $p1, p2, p3, p4, b_V$ and b_E are factors that may be trained and the CNN model is a multi-layer FC network. The advance process is described in Eq. (14) as follows:

$$CNN\left(i^l\right) = \sigma\left(W^{(l)} i^{(L-1)} + b^{(l)}\right) \quad (14)$$

Where, σ denotes the exponential activation operation and l stands for the layer depth. $CNN\left(i^l\right), W^{(l)}$ and $b^{(l)}$ are the resultant, weight and bias of the model correspondingly. In order for the GCN to acquire the KG, the compressed modules facilitate the transfer of complicated attributes as shared features.

3.6 Dynamic user preferences

The CNN and GCN features (individually represented for long and short preferences) are combined to create dynamic features, incorporating both user interest and items, which serve as input for the feature interactive layer, ensuring a comprehensive understanding of user preferences. The operation of attribute fusion for both CNN and GCN is represented in Eq. (15) as follows,

$$Z_i = W^p.(G_i \oplus \delta\, h_i) \quad (15)$$

In above Eq. (15), G_i and h_i denotes the attributes of the CNN and GCN models respectively. $W^{(p)}$ defines the weight matrix of the adjustable variables. $\oplus$ is the element-wise combinations and δ is a parameter that regulates the integrative proportion which is adjusted to 1. The softmax layer takes the CNN and GCN's dynamic (fused) information and utilizes them wisely for predictions. The depth recommendation loss $\mathcal{L}_{DREC}$ and the KG embedding loss $\mathcal{L}_{KGE}$ are components of the overall loss operation during training. Eq. (16) and Eq. (17) provides the expression for $\mathcal{L}_{DREC}$ and $\mathcal{L}_{KGE}$,

$$\mathcal{L}_{DREC} = MSE\,(Y, Y_{item}) \quad (16)$$

$$\mathcal{L}_{KGE} = Sigmoid\,(rt^T, \grave{p}t) \quad (17)$$

This raises the score for all relevant class and lowers the rating for all irrelevant class, where rt is the actual tail entity and pt is the entity predictive E. Following this, the following will be the optimization target (loss function) is mentioned in Eq. (18):

$$\mathcal{L} = \mathcal{L}_{DREC} + \mathcal{L}_{KGE} \quad (18)$$

This loss function is improving the training of classifier and the trained model is then utilized to predict test dataset. The label of the training dataset and the actual label of the datasets are used to calculate the loss of training. The preferences of the users are listed based on the loss value obtained for test data instances. Thus the preferences of items for the users are predicted then predicted items recommended for similar users.

4. Result and discussion

4.1 Dataset description

- **Amazon Digital Music Dataset [26]:** In order to assess the efficacy of recommendation systems for digital music products, this dataset includes 3,540 users, 3,568 items and 6,4706 reviews for items. So that the algorithm can efficiently recommend digital music products, the dataset includes detailed information about every review, such as the review text, rating and user details.
- **Book Crossing Dataset [27]:** This dataset includes a set of book ratings that includes 1,149,780 ratings for 271,379 books submitted by 278,858 people. Ratings for all books are on a scale from 1 to 10. Users' ratings for individual books are represented in each row of this dataset, which also includes book IDs and user ratings. Aside from the user ID, book ID, and rating, this data set also includes an extra set of time stamp information to record when the user rated the book.

4.2 Experimental setup and performance evaluation

This section evaluates the efficacy of the TAK-DepNR model by comparing it with other recommendation models like such as IGNN [15], AFMRUI [21], ICFRNN [23], DRR-Max [24] and SA-BiLSTM [25]. The implementation of both proposed and existing model is executed on a system with an Intel® CoreTM i5-4210 CPU @ 3GHz, 4GB RAM and a 1TB HDD running on Windows 10 64-bit which is carried out in Python 3.11 language for datasets illustrated in section 4.1. Table 2 presents the parameters and their values utilized for simulating both existing and proposed model to measure performance. The comparative evaluation is performed with respect to the user interest node (k) in terms of Precision, Recall and Mean Average Precision (MAP) to determine the efficiency of different recommendation model are listed below.

Table 2. List of optimal hyperparameters for both proposed and existing models

Models	Parameter	Range
IGNN [15]	Learning rate	0.001
	Activation function	ReLU
	Epochs	60
	Loss Function	Cross - Entropy
	Batch Size	50
	Optimizer	Adam
	Dropout	0.5
AFMRUI [21]	Embedding Vector	6
	Hidden Unit Number	50
	Learning rate	0.0001
	Activation function	ReLU
	Epochs	60
	Loss Function	Cross - Entropy
	Batch Size	512
	Optimizer	Adam
	Dropout	0.3
ICFRNN [23]	Vocabulary Size	20000
	Input Size	50
	Output size	100
	Learning rate	0.001
	Activation function	ReLU
	Epochs	60
	Batch Size	120
	Dropout	0.25
DRR-Max [24]	Learning rate	0.0001
	Activation function	ReLU
	Epochs	40
	Discount Rate	0.9
	Loss Function	Cross - Entropy
	Batch Size	64
	Optimizer	Adam
	Dropout	0.5
SA-BiLSTM [25]	Hidden size	256
	Learning rate	0.0001
	Activation function	Sigmoid
	Epochs	10
	Loss Function	Cross - Entropy
	Batch Size	32
	Optimizer	Adam
	Dropout	0.5
Proposed TAK-DepNR	Learning rate of recommended module	0.02
	Learning Rate of Knowledge Graph Encoding Learning	0.01
	Activation function	ReLU
	Epochs	80
	Weight Decay	9.59
	Momentum	0.9
	Loss Function	MSE
	Batch Size	1024
	Optimizer	Adam
	Dropout	0.5

Precision@k: Precision in Eq. (19) evaluates the percentage of suggested products that are relevant to the user. *Precision@k* calculates precision at a specific cutoff point K, which represents the number of top recommendations to consider.

$$Precision\ @k = \frac{Numer\ of\ relevant\ items\ in\ top\ k\ recommendations}{Number\ of\ items\ in\ top\ k\ recommendations} \quad (19)$$

Higher precision@K indicates better recommendations. Here k defines the user interest node.

Recall@k: It indicates the proportion of appropriate suggestions sent to the customers. *Recall@k* calculates recall in Eq. (20) at a specific cutoff point K. Higher recall@K indicates better coverage of relevant items in the recommendations.

$$Recall@k = \frac{Number\ of\ Relevant\ Items\ in\ top\ k\ recommendations}{Total\ number\ of\ relevant\ item\ for\ the\ user} \quad (20)$$

MAP(k): MAP in Eq. (21) measures the average precision across different cutoff points. It considers both precision and the position of relevant items in the recommendation list. In order to compute the $MAP(k)$, calculate the average precision $(AP)@k$ Eq. (22) for each user taking the average of the precision values at the relevant positions Eq. (23).

$$MAP@k = \frac{1}{number\ of\ users} \sum_{k=1}^{Number\ of\ users} AP@k \quad (21)$$

$$AP@k = \frac{1}{Number\ of\ relevant\ items} \sum_{k=1}^{k=n} Precision@k * relevant(k) \quad (22)$$

$$relevant(k) = \begin{cases} 1, & if\ item\ at\ k^{th}\ rank\ is\ relevant \\ 0, & otherwise \end{cases} \quad (23)$$

Fig. 2 and Fig. 3 show the $precision@k$ values for several recommender models using collected datasets based on various k node. It is observed that the precision value of proposed TAK-DepNR is greater than the other existing models on both the datasets. For the amazon digital music dataset with 20k, the precision of TAK-DepNR is 16.02% greater than IGNN, 10.70% greater than AFMRUI, 5.89% greater than ICFRNN, 3.58% greater than DRR-Max and 2.02% greater than SA-BiLSTM models. Similarly, for the book crossing dataset with 20k, the precision of TAK-DepNR is 17.37% greater than IGNN, 9.87% greater than AFMRUI, 7.31% greater than ICFRNN, 3.42% greater than DRR-Max and 1.39% greater than SA-BiLSTM models.

Fig. 4 and Fig. 5 show the recall@k values for several recommendation systems for the collected datasets based on different k nodes. It is observed that the recall value of proposed TAK-DepNR is greater than the other existing models on both the datasets. For the amazon digital music dataset with 20k, the recall of TAK-DepNR is 11.76% greater than IGNN, 7.73% greater than AFMRUI, 3.46% greater than ICFRNN, 2.98% greater than DRR-Max and 0.88% greater than SA-BiLSTM models. Similarly, for the book crossing dataset with 20k, the recall of TAK-DepNR is 8.54% greater than IGNN, 7.42% greater than AFMRUI, 3.66% greater than ICFRNN, 3.31% greater than DRR-Max and 0.42% greater than SA-BiLSTM models.

Fig. 6 and Fig. 7 show the $MAP@k$ values for multiple recommender methods using the collected datasets for diversed k nodes. It is observed that the MAP value of proposed TAK-DepNR is greater than the other models on both the datasets.

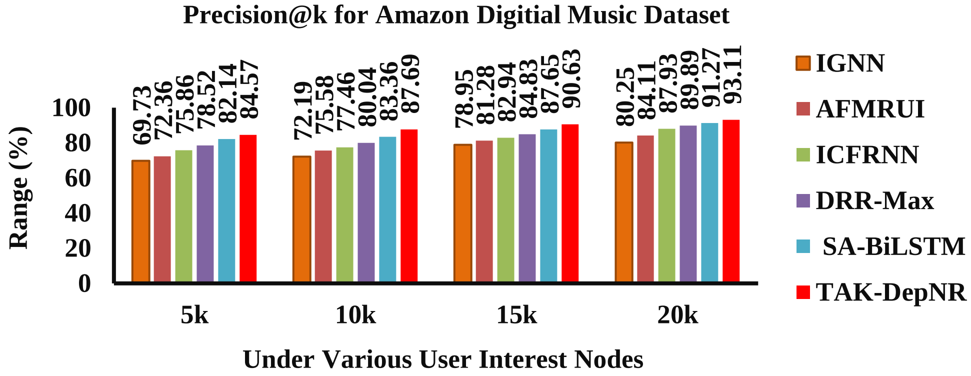

Figure. 2 Comparison of Precision@k for Amazon Digital Music Dataset

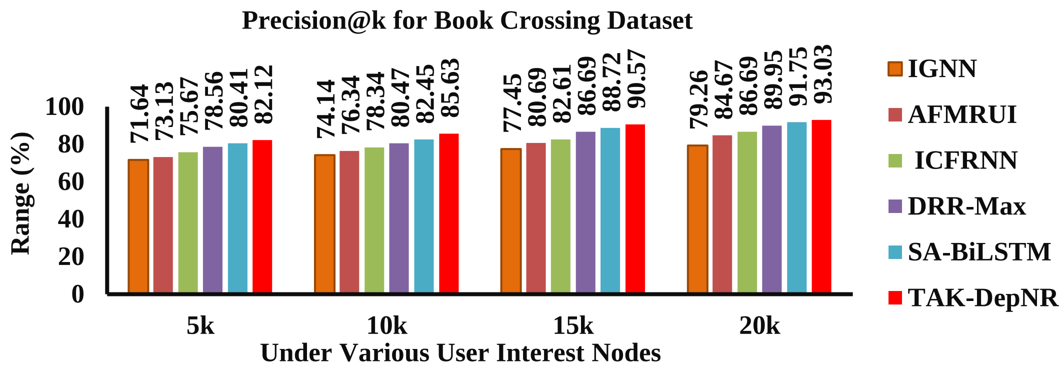

Figure. 3 Comparison of Precision@k for Book Crossing Dataset

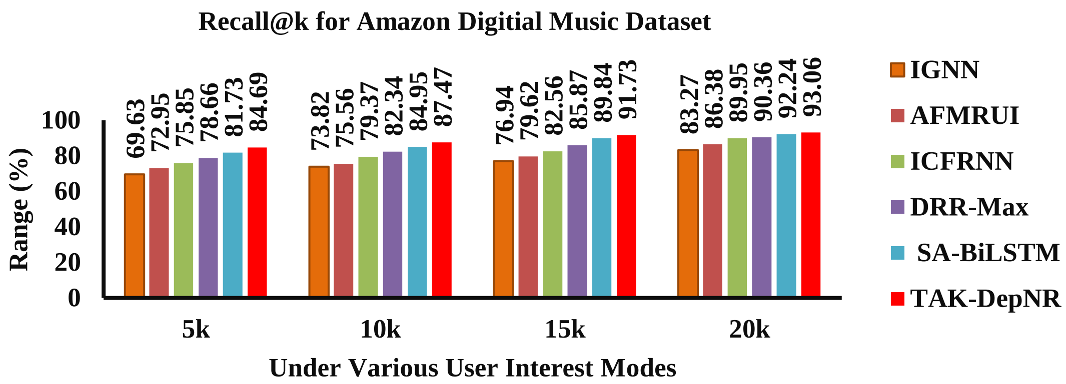

Figure. 4 Comparison of Recall@k for Amazon Digital Music Dataset

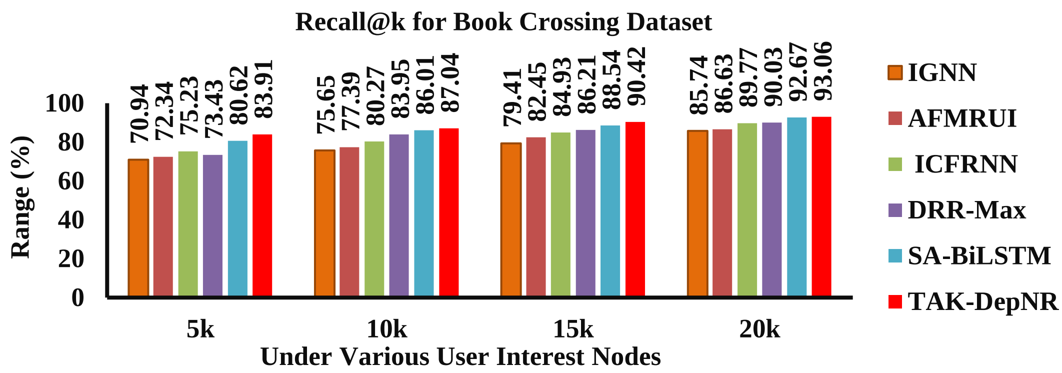

Figure. 5 Comparison of Recall@k for Book Crossing Dataset

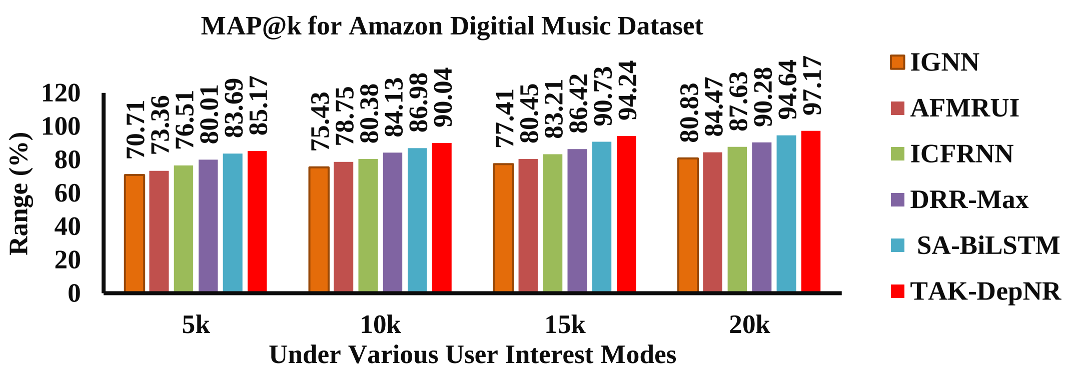

Figure. 6 Comparison of MAP@k for Amazon Digital Music Dataset

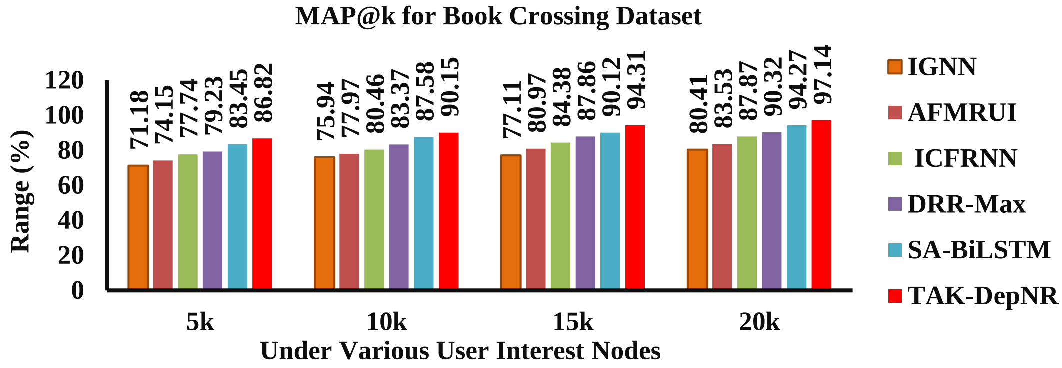

Figure. 7 Comparison of MAP@k for Book Crossing Dataset

For the amazon digital music dataset with 20k users, the MAP of TAK-DepNR is 20.21% greater than IGNN, 15.03% greater than AFMRUI, 10.89% greater than ICFRNN, 7.63% greater than DRR-Max and 2.67% greater than SA-BiLSTM models. Similarly, for the book crossing dataset with 20k, the MAP of TAK-DepNR is 20.81% greater than IGNN, 16.29% greater than AFMRUI, 10.55% greater than ICFRNN, 7.55% greater than DRR-Max and 3.04% greater than SA-BiLSTM models.

In the literature, IGNN [15], DRR-Max [24] models have utilized amazon digital music dataset. Similarly, AFMRUI [21], ICFRNN [23] and SA-BiLSTM [25] utilized book crossing dataset. Hence, this work evaluates proposed and existing models on both dataset using the parameters as per Table 2. From the above comparison, it is proved that the proposed TAK-DepNR model determines efficient results on both datasets under user interest node (k) for the user-item recommendation system. This is because the integration of CNN and GCN models are employed to address the challenges of missing KG and eliminates the computational costs for better user recommendation systems.

5. Conclusion

In this paper, TAK-DepNR proposed method that combines CNN and GCN to address the missing KG in customer recommendation systems. This method involves in modeling the user and item similarity using the MF model, which is then fed into a CNN and GCN model to form dynamic user preferences. CNN captures higher-order features between user and item, then reduces to lower-order features. Two changes are made to the CNN network from the VGG16 model, global average pooling layer is added to minimize spatial dimensions and the max-pooling layer is removed from the final convolution blocks. The GCN encodes the KGGCFM of the CNN, helping to learn the interpretation of each node and provide more detailed item characteristics in recommendations. The TAK-DepNR achieves average precision of 89% and 88% compared to other models like IGNN, AFMRUI, ICFRNN, DRR-Max and SA-BiLSTM on Amazon Digital Music Dataset and Book Crossing Dataset. However, the user behaviour sequence is varied over time, so dynamic representations of user item relationships required at different time steps/slices. In future work, an improved dynamic representation-learning model can be developed to handle dynamic behaviours.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

Conceptualization, methodology, software, validation, Sathya; formal analysis, investigation, Duraisamy; resources, data curation Latha Maheshwari; writing—original draft preparation, Sathya; writing—review and editing, Sathya; visualization, supervision, Duraisamy.

References

[1] B. Zhu, D. Guo, and L. Ren, "Consumer preference analysis based on text comments and ratings: A multi-attribute decision-making perspective", *Information & Management*, Vol. 59, No. 3, p. 103626, 2022.

[2] S. Jiang, X. Qian, T. Mei, and Y. Fu, "Personalized travel sequence recommendation on multi-source big social media", *IEEE Transactions on Big Data*, Vol. 2, No. 1, pp. 43-56, 2016.

[3] H. Ko, S. Lee, Y. Park, and A. Choi, "A survey of recommendation systems: recommendation

models, techniques, and application fields", *Electronics*, Vol. 11, No. 1, pp. 1-48, 2022.

[4] H. Al-Bashiri, M. A. Abdulgabber, A. Romli and F. Hujainah, "Collaborative filtering recommender system: overview and challenges", *Advanced Science Letters,* Vol. 23, No. 9, pp. 9045-9049, 2017.

[5] M. Jalili, S. Ahmadian, M. Izadi, P. Moradi, and M. Salehi, "Evaluating collaborative filtering recommender algorithms: a survey", *IEEE access*, Vol. 6, No. 1, pp. 74003-74024, 2018.

[6] D. Roy, and M. Dutta, "A systematic review and research perspective on recommender systems", *Journal of Big Data,* Vol. 9, No. 1, 2022.

[7] A. M. Turk, and A. Bilge, "Robustness analysis of multi-criteria collaborative filtering algorithms against shilling attacks", *Expert Systems with Applications*, Vol. 115, No.1, pp. 386-402, 2019.

[8] Q. Zhang, J. Lu, and Y. Jin, "Artificial intelligence in recommender systems", *Complex & Intelligent Systems*, Vol. 7, No. 6, pp. 439-457, 2021.

[9] G. Yatnalkar, H. S. Narman, and H. Malik, "An enhanced ride sharing model based on human characteristics and machine learning recommender system", *Procedia Computer Science*, Vol. 170, pp. 626-633, 2020.

[10] A. F. O. U. D. I., Yassine, L. A. Z. A. A. R.,Mohamed and M. Al Achhab, "Intelligent recommender system based on unsupervised machine learning and demographic attributes", *Simulation Modelling Practice and Theory,* Vol. 107 p. 102198, 2021.

[11] R. Wang, H. K. Cheng, Y. Jiang, and J. Lou, "TDCF: a two-stage deep learning based recommendation model", *Expert Systems with Applications,* Vol. 145, No. 12, pp. 1-10, 2019.

[12] S. Sharma, V. Rana, and V. Kumar, "Deep learning based semantic personalized recommendation system", *International Journal of Information Management Data Insights*, Vol. 1, No. 2, p. 100028, 2021.

[13] Q. Shambour, "A deep learning based algorithm for multi-criteria recommender systems", *Knowledge-Based Systems*, Vol. 211, No. 1, pp. 1-13, 2021.

[14] C. Wei, J. Qin, and Q. Ren, "A Ranking Recommendation Algorithm Based on Dynamic User Preference", *Sensors*, Vol. 22, No. 22, pp. 1-15, 2022.

[15] Q. Zheng, and Q. Ding, "Exploration of consumer preference based on deep learning neural network model in the immersive marketing environment", *Plos one*, Vol. 17, No. 5, p. e0268007, 2022.

[16] W. Zhu, Y. Xie, Q. Huang, Z. Zheng, X. Fang, Y. Huang, and W. Sun, "Graph Transformer Collaborative Filtering Method for Multi-Behavior Recommendations", *Mathematics*, Vol. 10, No. 16, pp. 1-14, 2022.

[17] N. M. Ali, A. Alshahrani, A. M. Alghamdi, and B. Novikov, "SmartTips: Online Products Recommendations System Based on Analyzing Customers Reviews", *Applied Sciences*, Vol. 12, No. 17, pp. 8823, 2022.

[18] L. Li, "Cross-border E-commerce intelligent information recommendation system based on deep learning", *Computational Intelligence and Neuroscience*, Vol. 2022, No. 1, pp. 1-11, 2022.

[19] S. Yuan, "Analysis of Consumer Behavior Data Based on Deep Neural Network Model", *Journal of Function Spaces*, Vol. 2022 No. 1 pp. 1-10, 2022.

[20] H. Liu, W. Wang, Y. Zhang, R. Gu and Y. Hao, "Neural matrix factorization recommendation for user preference prediction based on explicit and implicit feedback", *Computational Intelligence and Neuroscienc*e, Vol. 2022, No. 7, pp. 1-12, 2022.

[21] Z. Li, D. Jin and K. Yuan, "Attentional factorization machine with review-based user–item interaction for recommendation", *Scientific Reports*, Vol. 13, No. 1, pp. 1-17, 2023.

[22] C. Choudhary, I. Singh, and M. Kumar, "SARWAS: Deep ensemble learning techniques for sentiment based recommendation system", *Expert Systems with Applications*, Vol. 216, No. C, p. 119420, 2023.

[23] F. Roy and M. Hasan, "An Item–Item Collaborative Filtering Recommender System Based on Item Reviews: An Approach with Deep Learning", *Vietnam Journal of Computer Science (World Scientific)*, Vol. 10, No. 4, pp. 517–536 2023.

[24] Y. E. Hou, W. Gu, W. Dong, and L. Dang, "A Deep Reinforcement Learning Real-Time Recommendation Model Based on Long and Short-Term Preference", International *Journal of Computational Intelligence Systems*, Vol. 16, No. 1, pp. 1-14, 2023.

[25] I. Karabila, N. Darraz, A. El-Ansari, N. Alami, and M. El Mallahi, "Enhancing collaborative filtering-based recommender system using sentiment analysis", *Future Internet*, Vol. 15, No. 7, pp. 1-21, 2023.

[26] https://jmcauley.ucsd.edu/data/amazon

[27] https://www.kaggle.com/datasets/somnambwl/bookcrossing-dataset