Library Progress International
Vol.44 No.3, Jul-Dec 2024: P. 22741-22755

Print version ISSN 0970 1052
Online version ISSN 2320 317X

Original Article

Available online at www.bpasjournals.com

Enhancing User Experience with Session-Based Recommendation Systems Using Gated Graph Convolutional Networks

Somen Roy[1*], Jyothi Pillai[2], Ani Thomas[3], S. P Shukla[4], and G.C. Biswal[5]

[1]Dept. of Computer Application, BIT, Durg, India,
somenbsp2024@gmail.com
[2]Dept. of Computer Science and Engineering, BIT, Durg, India
jyothi.pillai@bitdurg.ac.in
[3]Dept. of Information Technology, BIT, Durg, India
ani. thomas@bitdurg.ac.in
[4]Dept. of Electrical Engineering, BIT, Durg, India
sp.shukla@bitdurg.ac.in
[5]Dept. of Electrical Engineering, BIT, Durg, India
gc.biswal@bitdurg.ac.in

How to cite this article: Somen Roy, Jyothi Pillai, Ani Thomas, S. P Shukla, G.C. Biswal (2024) Enhancing User Experience with Session-Based Recommendation Systems Using Gated Graph Convolutional Networks. *Library Progress International*, 44(3), 22741-22755.

Abstract:

A session-based recommender system (SBRS) focuses on user's interests depending on their browsing habits to make appropriate recommendations in a working session. Most of the existing work in growing research makes only use of the recently observed interactions on user- items. Session-based recommendation is the type where the recommender tries to suggest actions to users based on their previous anonymous session. In past works, a session is treated as a single time-point moving user models and not items models to recommend for. But they are insufficient to retrieve precise user vectors in sessions and intricate interactions of products. To achieve accurate item embedding and intricate transitions of items, we propose a novel method, called Session-Based Recommendation with Gated Graph Convolutional Neural Networks (SB-GGCNN). In this proposed method, global representations of temporal session sequences are modelled as graph-structured sequences through Gated Recurrent Units (GRUs) and GNNs to account for considerable item transitions. Furthermore, a Graph Convolutional Network (GCN) as a message passing network was applied to learn better representation of vectors in the global space. In this way, sequences are obtained: global preference of that user, current interest of that session, which is done by applying attention networks. The performance is verified utilizing extensive experiments on the SB-GGCNN model focused on e commerce datasets. Results indicated that the proposed model outperforms the best.

Keywords: Session, Recommender systems, embeddings, recurrent neural network, Graph convolution network,

1. Introduction

Recommender systems (RS) are decision assistance tools designed to assist users in identifying, a wide range of items that are relevant to them. These systems are commonly used in many contemporary online services, including media streaming and e-commerce websites, where they frequently provide significant financial value for their suppliers [1]. Traditional, RS were only intended to represent consumers' long-term preferences. On the other hand, the objective of session-based recommendation scenarios is to propose items while the user is still

buying or browsing. In these circumstances, short-term goals and user requirements must be inferred from the interactions of the current session, and they frequently take precedence over long-term preferences [3,4]. In reality, it is extremely difficult to provide reliable session-based suggestions with only a small number of recently observed interactions. For instance, in e-commerce platforms, a large number of store visitors may be completely anonymous due to either the fact that they are not registered or are new customers. In these scenarios, there is absolutely no long-term preference data available. However, in a particular situation, when the client is already recognized and logged in, the user's short-term shopping objectives may vary from session to session. So, it is crucial to provide recommendations that align with those purposes [3, 5]. Therefore, the user's primary goal is to create global and local RNN recommenders and concurrently record the sequential behaviour of a user [7].
The aforementioned techniques discussed in [1, 3, 4, 5, 7] are state-of-the-art models and produce good results, but they nevertheless have certain drawbacks. First off, these approaches struggle to estimate user representations in the absence of sufficient user behaviour in a single session. Typically, the user representations are obtained from the hidden vectors. However, with session-based RS, user behaviour is involved in session clicks. Further, these sessions are typically anonymous and consisting several sessions as a result, estimating each user's representation from each session with accuracy is challenging. Second, prior research indicates that item transition patterns are significant and can be incorporated as a local factor in session-based recommendation [6, 7]. However, these approaches never account for transitions between contexts, or other items in the session, instead only model one-way transitions between consecutive items. Therefore, these approaches frequently miss intricate transitions between far-off objects. In recent years Graph neural network (GNN) has been utilized more efficiently to calculate the importance of each item the last item is crucial during the learning of session representation, GNNs as message-passing networks extract useful information where GCN, Graph SAGE and GAT are most frequently used graph neural network for real world applications.

To go beyond the previously stated constraints, the goal of this study is to represent multiple levels of session item conversion interactions using a graph neural network, which has the inherent ability to collect and train a wide range of complicated association relationships from graph data. In order to understand the interaction relationship between items from various levels, we are constructing global graph, session graph and feature graph on the basis of all session sequences. Then, we leverage a graph neural network's potent graph data learning and representation capabilities to obtain the complex relationship of items in a session. The next-click item can be deduced using the suggested SB-GGCNN, which creates more dependable session representations.

The main contribution of our work is as follows:

- We develop the SB-GGCNN model to address the global representation of session items with current interest representation and obtain complex relationships of items.
- We build a session graph and utilize the GRU based Graph neural network to obtain nodes' interaction order as weighted information in the graph's connection matrix. Also, incorporate the global neighbour graph to obtain global item interaction from the session.
- Moreover, the proposed model is further improved by employing a Graph Convolutional network (GCN) which acts as an aggregation or message-passing network over a global graph
- We conduct several experiments to validate the performance of the model on real-world datasets and tune the parameters to achieve better results. From the experimental outcomes, it is found that the SB-GGCNN model outperforms the baselines.

The rest of the paper is divided into several sections. The "Related work" section explores past research and common areas where session-based recommendation approaches are used. Following this, the proposed architecture methodology section discusses the technical aspects of the SB-GGCNN model. The "Experiments" section outlines the evaluation setup and performance of the model, while the results section summarizes the comparative analysis among the proposed and state-of-the-art techniques. At last, we discuss the conclusion with future scope.

2. Related Work

In this Section, we discussed some recent past work that are closely related to our work. To carry out session suggestions, traditional approaches often employ data mining or machine learning techniques to mine the relevance of the session data [7]. This type of recommendation approach uses common patterns and association rules for recommendation. Generally, traditional approaches start with data mining of the patterns, rules, and session matching, and end with recommended outcomes [8]. In the case of sequence pattern mining-based session recommendation algorithms, they utilise weighted sequence patterns with respect to a user for recommendations. That user's historical chain of interactions is compared, and based on category similarity the weight for each session. Moreover, Sequence Pattern Mining based models for short- and long-term preference of the user [10, 11]. To extract sequential patterns in sessions, NARM [12] Deep learning is more successful at extracting data features than standard approaches. Recommendation algorithms, which rely on deep learning, employ neural network models to train user interest preference and then utilize that representation to anticipate the next interaction [13–15]. Balazs Hidasi, for example, introduced the GRU4Rec model built on Gated Recurrent Unit (GRU) [15], based on processing of item sequence that is clicked or viewed by users within a session and uses only past interaction data to recommend. Deep Neural Network gives even more recourse to the session-based recommender system which demonstrated exceptional effectiveness in predicting the things that users would click next. A simple method for managing a session is to sort the order of objects the user clicked by time, and this was the first use of recurrent neural networks (RNNs) or attention mechanism models for extracting sequential patterns [12,13]. used the gated recurrent unit (GRU) [14], which is an extended version of RNN. SRSAN [13] a self-attention model exploits the relationship between items in the session. In contrast to RNN, GNNs are good at modelling item dependencies and session consistency, due to high performance [16–20], recent research has combined GNNs with other neural networks [15] to build a session-based recommender system. The original GNN-based framework, SRGNN [15] has served as a foundational framework for several investigations. The latent representation of nodes was obtained by sending a session graph into a gated GNN [21]. A soft-attention method then constructed the session representation using the node representations. Normalized representations of items and sessions produced by GNNs were used by Gupta et al. [17] to improve overall recommendation results. A unique GNN model was presented by Chen et al. [18] that focuses on preserving item ordering in sessions while aggregating the information supplied by neighboring nodes using a GRU. The Edge-Order Preserving Aggregation (EOPA) layer minimizes information loss issues and has a promising performance. TAGNN++ [19] enhanced the expressiveness of the attention model by adaptively activating it in response to a variety of target objects. LIU introduced the Short-Term Attention/Memory Priority (STAMP) model [20] in response to many researchers trying to accurately express a user's preference during his brief session, acquiring long-term interests of users from their own memories and visualizing current interest by last interacted item, which is effective on modelling the target behaviour.

Among different deep representation learning methods, graph neural networks (GNNs) have emerged as particularly suitable for representing graphs these years due to their inductive transfer capabilities and model interpretability [23–25]. Finally, GNNs can naturally capture the collaborative signal required to improve user and item representations together in an open manner. For example, a GNN aims to utilize network structure for, e.g., showing neighbouring node information (in some companies with. Then, the resultant node representations capture similarities and commonalities between the current node and its neighbours such as low-frequency information of nodes. These parallels and commonalities can apply to a majority of recommendation jobs. Numerous works have made outstanding achievements in RS with GNNs in the last few years. First off, in order to learn the information embedding vectors of each interaction node, we generate session sequence data as graph-structured data. They then use GNNs to encode the complex node-to-node transition dynamics in the session into embedding vectors. The SR-GNN model, Shu et al. therefore a gated graph neural network [28] is proposed, which outperforms former non-GNN models notably and meanwhile introduces soft attention on calculating session representation to represent user's long-term preference and short-term preference over item.

A gated graph neural network extract global characteristic of the session (long-term interest), and a self-attention network captures local features in those sessions reflecting short-term interests by GCSAN model [29]. The target item nodes of different items, obtained by the TAGNN model based on multiple candidate target items at each user time slice are used to construct a tensor containing the representation, so as to represent and capture how users' interests change with other target entities. Finally, it learns item initial embedding using gated graph neural network [30]. Pan et al., employed a star graph neural network and other elements to introduce the SGNN-HN model [31]. According to Wang et al. Based on Modgil and Scott's proposal, the GCE-GNN model constructs two kinds of graphs which are session graph and global graph respectively. It learns the item's global level item representation in the global graph, as well as its session-level item representation in the session graph. also, the Graph Convolutional Network is more effective in graph structure data for node classification [33]. Ultimately, from the above literature, we found, that the researchers did not use neighbor nodes and their edge information using Graph Convolutional Network (GCN) as a global domain to represent the global neighbor graph and obtain more accurate information. Therefore, we proposed a novel technique called SB-GGCNN to capture neighbor nodes and their information to represent the global domain for enhancing the performance of the model.

3. Proposed Model: Session-Based Gated Graph Convolutional Neural Network (SB-GGCNN)

In this section, we explain in more detail our model's suggested architecture to address the approach for predicting users' next actions. The proposed model consists of four parts such as (i) Representation learning of items at the session layer, (ii) Representation learning of items at the global layer, (iii) Aggregation layer, and (iv) Soft Max prediction scoring layer. The details of these parts are discussed in the following subsections. The framework of the suggested model is depicted in Figure 1

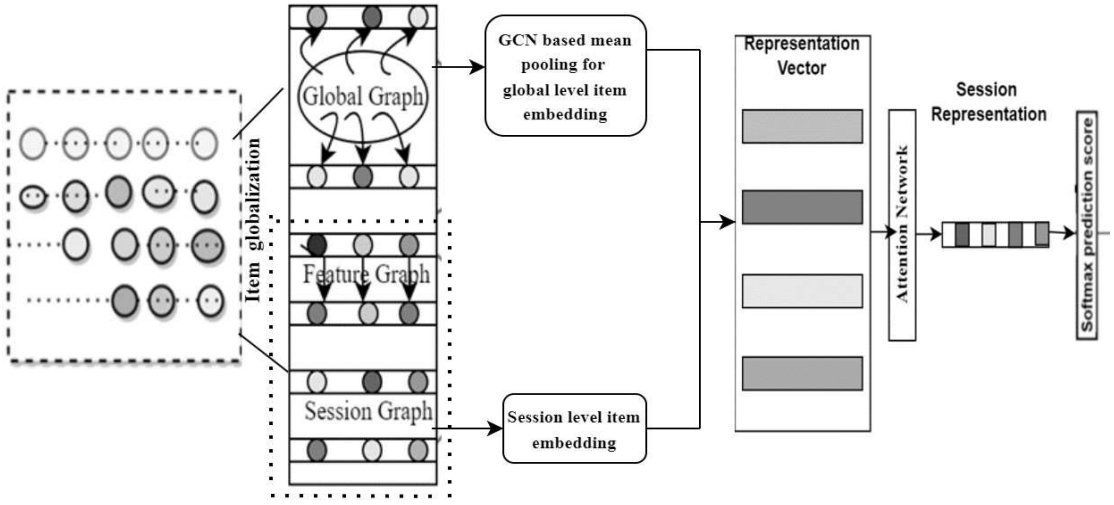

Fig. 1 The Architecture of the proposed SB-GGCNN model.

3.1 Representation Learning of Item for Session Layer

Building the Session Graph: $G_s = (V_s, E_s)$ represents a weighted directed graph. The current session sequence's item transition connection allows us to obtain the session-level item representation $S_i=(S_{i1}, S_{i2}, \ldots . S_{it})$. All items participating in all sessions are represented by the vertex V_s, and all item transition relationships in the session S that exist in V_s are represented by the edge set E_s. The edge (V_i^{t-1}, V_i^{t+1}) shows that the user clicked on an item V_i^{t-1} at time t in the session, S_i then clicked on the item at theV_i^{t+1} subsequent time step. Given that the goal of session suggestion is to suggest the current session's next interaction item, the final interaction item usually influences the subsequent interaction more. Consequently, the node's effect on the session suggestion increases with the proximity of the node to the conclusion of the session. Therefore, while building the session graph, we

take into account assigning varying weights based on the interaction order position of the nodes in the session and using the interaction sequence of the nodes as the weight information in the connection matrix. As an illustration, the session $S_1 = \{v_1, v_2, v_3, v_4, v_3, v_1\}$, as seen in Figure 2.

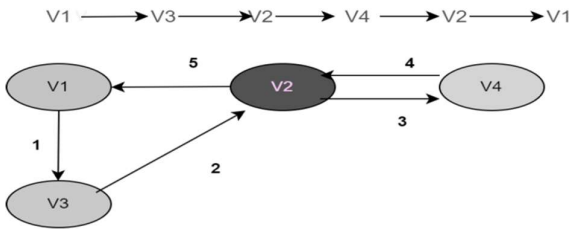

Fig. 2 Construction of the session graph for sessionS_1.

The following illustrates how the weighted session graph produced previously can be used to determine the session graph's connection matrix M_1.in Table 1

Table 1 The initial connection matrix M_1 of the session

	Outgoing edges				Incoming edges			
	1	**2**	**3**	**4**	**1**	**2**	**3**	**4**
1	**0**	**0**	**1**	**0**	**0**	**5**	**0**	**0**
2	**5**	**0**	**0**	**3**	**0**	**0**	**2**	**4**
3	**0**	**2**	**0**	**0**	**1**	**2**	**0**	**0**
4	**0**	**4**	**0**	**0**	**0**	**3**	**0**	**0**

We use an equation to normalize the weight information because the lengths of the actual session data vary define in equation (1)

$$weight(y) = Means \times \left(\frac{1}{1+e^{-\frac{y}{2}}} \right) \quad (1)$$

Weight (y) provides the weight information in the connection matrix M_1, and the average session length of the session sequence data is represented by y ϵ {1, 2, 3, 4... n}, Means. The weight information formula above states that the final output weight size is mapped by (0, Means). Further a standardized connection matrix M_1^t Obtained for session 1 and propagated to GNN along with parameter information. We use the item representation vector and the item feature information representation vector are encoded using a graph neural network according to equation (2).

$$A^t{}_{si} = M_1^t[V_1^{t-1} \dots\dots\dots V_n^{t-1}]^T \; U + b \quad (2)$$

Next, a gated graph neural network is deployed, in which a message-passing model is based on a gated recurrent unit, specifically designed to work with sequential problems. We use the connection matrix. M_1^t to acquire each node's information in GGNN and enhance the nodes' information propagation process. After that, the particular learning process takes place. as per Equations (3) to (6).

$$P^t{}_{si} = \sigma \; (W_z A^t{}_{si} + X_z V_1^{t-1}) \quad (3)$$

$$Q^t{}_{si} = \sigma \; (W_Q A^t{}_{si} + X_Q V_1^{t-1}) \quad (4)$$

$$\check{\mathrm{R}}_{vi}^{t} = tanh\,(W_o A^{t}{}_{si} + X_o(Q_{si}^{t} \bullet V_s^{t-1})) \tag{5}$$

$$R_{vi}^{t} = (1 - P_{si}^{t}) \bullet V_i^{t-1} + P_{si}^{t} \bullet \check{\mathrm{R}}_i^{t} \tag{6}$$

Where $A^{t}{}_{si}$represents item hidden vector, while M_1^t Is the connection matrix $P^{t}{}_{si}$and $Q^{t}{}_{si}$Are Each session, the reset and update gates in the gated graph neural network separate the hidden vector into information that is valuable and information that is useless. Ultimately, we use the sigmoid activation function to determine whether to keep or discard, Where (•) represents multiplying components one by one. To determine the candidate state, we utilize the update gate and the $tanh$ function. To acquire the vector representation R_{vi}^{t}Of each node at time t. To obtain item representation feature vector for a session using Equations (7) and (8). Also, a reset gate is employed for item feature representation vector-related session node R_{fi}^{t} Using Equations (9) and (10).

$$P^{t}{}_{si} = \sigma\ (W_z A^{t}{}_{si} + X_z R_{t-1}^{f}) \tag{7}$$

$$Q^{t}{}_{si} = \sigma\ (W_Q A^{t}{}_{si} + X_Q R_{t-1}^{f}) \tag{8}$$

$$\check{\mathrm{R}}_{t}^{f} = tanh\,(W_o A^{t}{}_{si} + X_o\big(Q_t^{t} \bullet R_{t-1}^{f}\big)) \tag{9}$$

$$R_{fi}^{t} = (1 - P_{si}^{t}) \bullet R_{fi}^{t-1} + P_{si}^{t} \bullet \check{\mathrm{R}}_{fi}^{t} \tag{10}$$

We concatenate the $\check{\mathrm{R}}_{vi}^{t}$ and R_{fi}^{t} to find the final session-level item output vector R_{vi}. For each session graph and item feature graph, the gated graph neural network simultaneously updates its nodes. $A^{t}{}_{si}$ Is utilized to extract latent vectors from neighbourhoods, feed them into a graph neural network as input data, and then apply update gates to facilitate information transmission between various nodes within the limitations provided by the item connection matrix and reset gates and decide which information to retain or discard as defined in Equation (11).

$$R_{vi} = Concat\,(\hat{R}_{vi}^{t}, R_{ft}^{t}) \tag{11}$$

3.2 Representation Learning of Items for the Global Graph

The final purpose of the global graph is to choose out good information from the global sequence, which does not contain any session noise. we use the following equally averaging formula to calculate this for current node and neighbourhood node on neighbourhoods' graph; This is basically the use of a Graph Convolutional Network (GCN) to propagate message from node to corresponding nodes and weight different items with different degrees in the global graph, virtualizing them as the current item. We employ the graph neural networks to learn from such features of the item and then inspect how deeply we can go into understanding these preferences inside it. It can be seen that our method is significantly superior to the other trainable pooling techniques Our approach models the spatial relation using the gated recurrent unit (GRU) for higher-order neighbour information in addition to capturing the first-order neighbour's information by learning attention scores. The relevance of R_{vi}^{l+1} is defined in Equation (12).

$$R_{vi}^{l+1} = Relu\left(\sum_{j \in ni} R_{vj}^{l} \frac{W^{I}}{\sqrt{|n_i||n_j|}}\right), \qquad v_i \in V \tag{12}$$

Where R_{vi}The representation is a vector of the node.$v_i$and W^I Is the learnable parameter weight matrix, $\sqrt{|n_i||n_j|}$ is the normalization constant is determined by the graph structure. The vector representations of each neighbour node were then linearly merged based on the relevance probability and vector representation for neighbor node is obtain by Equation (13).

$$R_{N_a(vi)} = \sum_{vi \in N_a(vi)} W\big(vi, v_j\big)\, R_{vi} \tag{13}$$

To implement the aggregation function and perform the weight transformation or message passing, we now aggregate the current node vector. R_{vi} and the vector representation of its global domain node $R_{N_a(vi)}$ Is defined in Equation (14).

$$R_{vi}^{g} = Relu\ (W_2[R_{vi}\ ||R_{n_a(vi)}]) \tag{14}$$

Further, a convolution aggregator is utilized to combine more relevant data and carry out several aggregations about the target items and this representation of items in the k[th] step is defined in Equation (15)

$$R_{vi}^{g(k)} = agg\left(R_{vi}^{(k-1)}, R_{N_{vi}}^{n(k-1)}\right) \tag{15}$$

The item represented by the preceding term via the information propagation process is $R_{vi}^{g(k-1)}$Its original value the item's K[th] order aggregated representation is derived from integrating its initial embedding representation and the embedding representation of all of its global graph neighbor's. This process produces a more efficient final item representation, which is ultimately based on the item feature vector. R_{fi} That was previously found. Now we concatenate $R_{vi}^{g(k)}$ and R_{fi} To acquire the item output vector's final global domain layer in accordance to Equation (16).

$$R_{vi}^{g} = Concat(R_{fi}, R_{vi}^{g(k)}) \tag{16}$$

3.3 Aggregation Phase

This component combines the session graph's vector representation with the global graph to produce the node's final form and, in the end, the conversation's sequential representation. A neighbourhood aggregation or message-passing scheme is typically used to generalize the convolution function to irregular graph domains. This automatically handles message propagation; facilitating the creation of such message-passing graph neural networks that utilize a straightforward message-passing function to this proposed model. We will rely on an aggregation process using the "mean" convolution operators and presume that the global neighborhood graph and the session graph's vector representation are equally significant. The final vector representation for every node is the combination of session-level vector representation and global domain-level representation yields. R_{vi}^{t} As defined in Equation (17).

$$R_{vi}^{t} = (R_{vi} +, R_{vi}^{g}) \tag{17}$$

Likewise, to get the feature representation vector of the conversation, we take the average of the item representation vectors of each node in the conversational sequence that the GNN produces. **as defined in Equation (18)**

$$S^{t} = \frac{1}{l}\sum_{i=1}^{l} R_{v^{s_i}}^{t} \tag{18}$$

To determine each node's contribution weight to the conversational sequence representation, we employ soft attention. **according to Equation (19).**

$$T_i = Q_2^t\ \sigma\ (W_4 R_{vi} + W_5 S^t + b_4) \tag{19}$$

Where $T_i$represents soft attention and W_4, W_5, b_4,ϵ R^d are the trainable parameter,

Lastly, we may acquire the conversation's final interest representation vector by utilizing the contribution weight of each node, which is computed to linearly combine item representation.as follows by equation (20)

$$U_s = \sum_{i=1}^{l} T_i R_{v^s i}^t \tag{20}$$

In the method described above, the global domain layer and session layer node representations are integrated by U_s and T_i Stands for soft attention. Lastly, the $R_{v^s i}^t$ Indicates Using the session sequence, the item representation vector of each item in the chat was fed into the GNN. The global graph construction process not only considers the frequency of each edge in each conversation, but we also use the graph neural network to fully mine the item preference hidden in the item feature information, better model the relationship between items in the conversational data, and efficiently utilize the conversational information to better predict the user's interest preference.

3.4 Prediction Phase of the model

Based on the beginning vector of each item and the final interest representation vector of session S, we may use it in this section to generate predictive suggestions. The predictive score y of the subsequent item is then obtained. $v_i \in V$ can be obtained according to Equation (21)

$$\hat{y}_i = softmax(U_s^T R_{vi}) \tag{21}$$

$\hat{y}_i$ represents the probability distribution of all suggested items, and since the SoftMax function also computes the input predicted scores, we opt for the cross-entropy loss function. defined in Equation (22) as the loss function.

$$l = -[y log\ \hat{y} + (1 - y)\log(1 - \hat{y})] \tag{22}$$

Here y is the actual score of the item and $\hat{y}$ Is the predictive score of the item, if y = 1 then l = - $log\ \hat{y}$ The loss function replicates the difference between the expected and actual scores, as the projected score likewise tends to be 1. The Final loss function is defined in Equation (23).

$$l = -\sum_{i=1}^{n} y log(\hat{y}) + (1 - y)\log(1 - \hat{y}) + \frac{\lambda}{2}||W||^2 \tag{23}$$

Where y is the actual score of the item, and $\hat{y}$ Is the predicted score obtained, and λ is employed as a regularization factor to avoid overfitting, and W the set of all learnable parameters, is a coefficient that balances the two terms of the loss function.

4 Experimental Setup and Dataset

In this part, we briefly discussed the experimental setup and dataset that was used to validate the experiments and model. In the experimental setup, we used the following software Torch Geometric, Google Colab, Kaggle Notebook, and Windows 11. Similarly, hardware GPU and Intel core i7 are used.

4.1 Data Set

We have utilized the **Retail Rocket Dataset**[1] **[34]. This is real-world data collected from an e-commerce online platform for building session-based recommendations** The dataset has the following features like number of clicks that were collected over 4.5 months. Users can create three different kinds of events: "view," "add to cart," and "transaction." A total of 2,756,101 events, comprising 2,664,312 "views," 69,332 "add to cart," and 22,457 "transactions," were created by 1,407,580 distinct users. The item properties file contains

[1]https://www.kaggle.com/retailrocket/ecommerce-dataset

20,275,902 rows, or various properties, that describe 417,053 unique items. Table 2 displays the dataset's detailed statistics.

Table 2 shows the detailed statistics of the dataset.

Dataset	Retail Rocket [34]
Clicks	266,4312
No. of the session in the training set	373,755
No. of the session in the test set	52,584
No. of Items	446,867
Average session length	< 5

4.2. Evaluation matrices

Two popular ranking-based evaluation indicators, HR@ K (hit ratio) and MRR@N (mean reciprocal rank), are used to support the model's performance. where HR@K is the number of successfully hit items from the top N items; the higher HR, the greater the model's recommendation effect. Equation (24), which defines the hit ratio.

$$Hit\ Rate = \frac{|U_{hit}^{l}|}{|U_{all}|} \quad \textbf{(24)}$$

Where U_{hit}^{l} The Top@ N suggestion list includes the number of users for whom the appropriate recommendation is present. U_{all} **is the** the test dataset's total number of users

The accuracy of recommendations while taking ranks into account is also measured by Mean Reciprocal Rank (MRR). Equation (25), which defines MRR, is the mean of the reciprocal rank of ground truths in the lists.

$$MRR = \frac{1}{|S|}\sum_{i=1}^{|S|}\frac{1}{rank_i} \times 100 \quad (25)$$

4.3. Baseline Approaches

Compare the performance of the suggested approach with the following benchmarks. These fall into one of three categories. Attention-based techniques, based method and the GNN-based approach

- NARM [12]: When making suggestions, GRU4Rec uses an attention mechanism to ascertain the user's major interest. This approach is an enhanced model that is based on the user's primary aim for the current session, which is presented first.
- SR-SAN [13] this technique substitutes a self-attention-based mechanism for the attention-based mechanism of SRGNN. The self-attention-based mechanism adaptively captures the link between items' transitions and interactions, gaining the user's sustained attention.
- GRU4REC [15]: This approach is an RNN-based approach that models user sequences using gated recurrent units (GRU). It makes use of the item interaction data across the whole series and employs a ranking-based loss function.
- SR-GNN [16]: In this method, Soft attention is used to acquire the session representation, and a gated graph neural network represents each item. It symbolizes the user's sustained interest. The final item in the sequence indicates the user's short-term interest, although it also uses long-term and short-term interest.
- NISER [17]: The normalization of the item and session-graph representations is how this method improves overall recommendation performance above the state-of-the-art outcomes.

- EOPA [18]: The EOPA and SGAT layers in this model are based on the S2MG and S2SG conversion Algorithms, which turn sessions into graphs. A model known as LESSR that eliminates the two information loss issues is created by merging the two types of layers.
- TAGNN++ [19]: This model enhances SRGNN by creating a target-aware attention mechanism. This model generates a representation of the target item for every node vector of every item acquired therefore Representing the user's fluctuating interest concerning distinct target items.

4.4. Parameter setting

On a validation set, which is a 10% random subset of the training set, we choose additional hyper-parameters. A Gaussian distribution with a mean of 0 and a standard deviation of 0.1 is used to initialize all parameters. The model has 30 epochs, and the mini-batch Adam optimizer is used to optimize these parameters. The weight decays by 0.1 every 5 epochs, and the starting learning rate is set at 0.001. Additionally, the L2 penalty is set at 10^{-4} and the batch size is set at 128.

5. Experimental Result Analysis

In this section, we compare our model to the most advanced models on real-world datasets in order to validate it. Furthermore, the original data was divided into a train and test set at an 80:20 ratio. Furthermore, we compare the suggested method with seven baseline approaches using two evaluation measures, such as mean reciprocal rank (MRR) and hit rate (HR), and find that our proposed SB-GGCNN performs better than baseline approaches. The experimental results are compared in Table 3.

Table 3 Experimental outcomes on retail rocket datasets are compared.

Dataset	**Retail Rocket**			
Metrics	HR@10	MRR@10	HR@20	MRR@20
NARM	39.02	21.07	45.64	21.53
EOPA	33.63	17.33	42.28	17.91
NISER	33.33	22.09	37.66	22.39
SR-GNN	34.40	21.92	39.45	22.27
SR-SAN	37.50	17.26	44.33	17.75
TAGNN	34.10	21.71	39.61	22.09
GRU4REC	41.50	23.80	48.00	**24.30**
Proposed SB-GGCNN	**43.51**	**21.95**	**52.12**	**22.57**

Experimental outcomes show the performance of the baseline models with our proposed SB-GGCNN. We evaluate the performance in both top@10 and @20 recommendation list on the Retail Rocket dataset. It can be noted from Table 3, that the proposed (SB-GGCNN) model shows better performance with second best model in HR @10 metrics with an improvement of 2.01% to second best model (GRU4REC) similarly our model improves 4.12% of HR@20 compared to the second-best model. Further, we found that our proposed model is better than the attention-based (NARM, SR-SAN) model with an improvement of 4.49 % and 6.01 % respectively in HR@10. Similarly, our model improves by 6.48 % and 7.79% by HR@20 in comparison to the NARM and SR-SAN models respectively. Further, it is found that SB-GGNN model is better than GNN-based models (EOPA, NISER, SR-GNN, and TAGNN) with an improvement of 9.88 %, 10.18 %, 9.11 %, 9.41% in HR@10 and 9.74 %, 14.46%, 12.67 %, and 12.51 in HR@20 respectively. The outcomes signify that the SB-GGNN model outperforms the baseline approaches with Hit Rate evaluation metrics.

On the other hand, it can be seen that the proposed model performance with MRR@ K is better than the baseline model NARM, EOPA, SR-SAN, and TAGNN with improvements of .88 %, 4.62 %, 4.69 % and .24% for MRR@

10 and performance with SR-GNN model is almost similar with improvements of 0.03%.Where is the performance of MRR@10 not overcome to the performance of NISER and GRU4REC models Similarly, the performance of MRR@20 is better than the baseline models, NARM, EOPA, NISER, SR-SAN, SR-GNN, and TAGNN with improvements of 1.04%, 4.66%, 0.18%, 0.30%, 4.82%, 0.48%, respectively whereas, the performance of the MRR@20 is not overcome the performance of GRU4REC model. The above experimental results in Table 3 indicate that our Proposed SB-GGCNN, outperform all the state-of-art baselines consistently by HR@K metrics whereas in the case of MRR@ k, its performance seems to be the second-best model. Which overcomes all the baselines except GRU4REC.

In order to visualize the data, Figures 3 and 4 compare the performance of each model on HR@10 and HR@20 for the Retail Rocket dataset. It is evident that the SB-GGCNN model that was suggested performs the best out of all the state-of-the-art models.

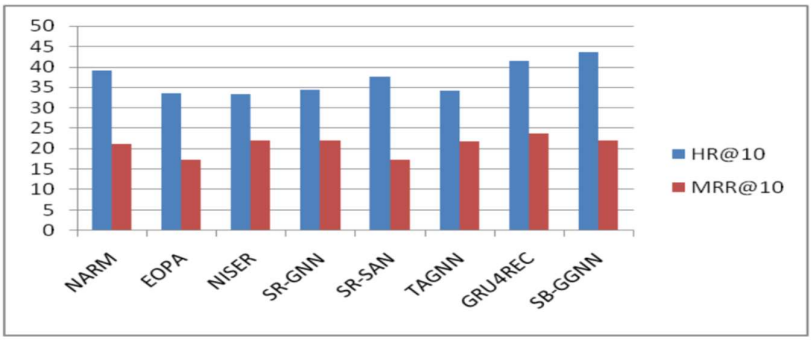

Fig.3 Comparison of model's performance for HR@10 and MRR@10

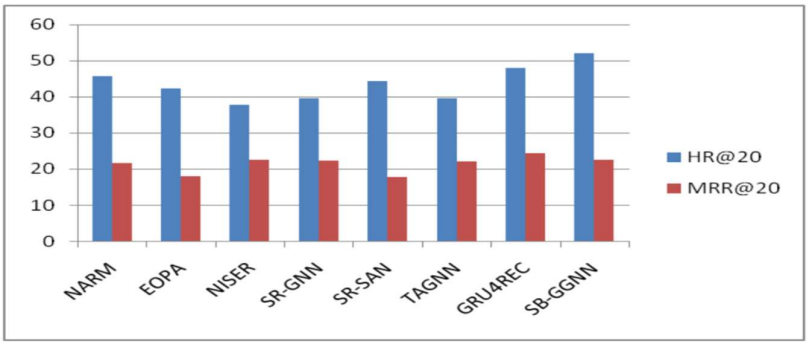

Fig. 4 Comparison of model's performance for HR@20 and MRR@20

Performance of SB-GGCNN model accuracy in terms of training and testing curve as shown in Figure 5, Shows that the testing accuracy steadily increases with the progress of epoch from 1 to 20 where testing accuracy neither increases nor decrees for epoch 20 to 30 similarly in the case of training lose epoch from 1 to 20 is decrees and training lose remains the study state from epoch 20 to 30.

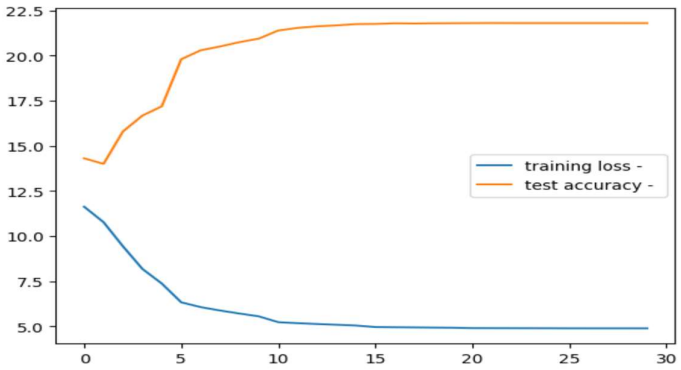

Fig. 5 Model training loss and test accuracy

5.1 **The Impact of Session Length with Mean Convolutional Operator**

In this Section, we studied the impact of the session length of the SB-GGCNN model. Figure 6 shows the impact of the hit rate with respect to the session length. From Figure 6, it is noted that the model performance steadily increases from session length 1 to 2 whereas the session length increases exceptionally from 2 to 5. However, the performance suddenly decreases when the session length proceeds from 5 onwards. From the above result, we assume that the model is over-fit after session length 5, the comparative experimental outcomes for HR@ 20 as follows in table 4.

Table 4 comparison of various session length with different weight decay for proposed model

Proposed SB-GGCNN	Weight decay (0.1)	Weight decay (0.2)	Weight decay (0.3)	Weight decay (0.4)
Session length 1	49.50	49.55	49.65	49.61
session length 2	49.77	49.91	50.15	50.36
session length 3	50.61	51.11	51.45	51.46
session length 4	51.81.	52.01	52.12	52.11
session length 5	**52.12**	**52.16**	**52.15**	**52.12**
session length 6	51.10	51.15	51.22	51.21

To visualize the data, Figures 6 show the comparison performance of proposed models in mean convolutional pooling adopted by Graph Convolutional Network (GCN) on various session length (1 to 6) in terms of different weight decay (0.1 to 0.4) for the Retail Rocket dataset it is clearly shown that the performance of the proposed SB-GGCNN model is best when the session length is between 3 to 5 for HR@ 20.

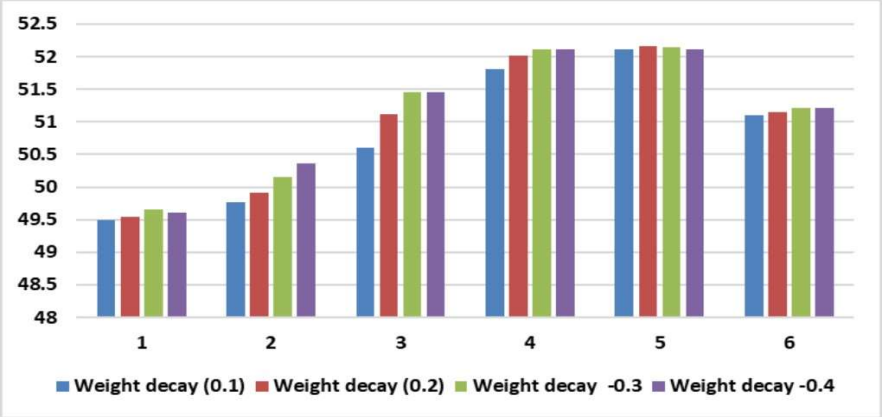

Fig. 6 The impact of session length with different weight decay on model performance

6. Conclusion and Future Work

This work introduces SB-GGCNN, a unique session-based recommendation architecture that uses graph models to represent session sequences. The suggested approach creates a plan to discover higher-order relationships between node elements in addition to taking into account their intricate structure and transitions. Further, the SB-GGCNN obtains hidden information about all neighbor nodes to extract more useful information for the session. In addition to this, the model applies convolutional based aggregation scheme to aggregate the message passing with the help of a gated recurrent unit and graph convolution network in the global domain, which is more efficient for obtaining better information. Also, a soft attention mechanism is applied to predict the user's current interest representation with the next click action. The comprehensive experiment confirms that the proposed model consistently outperforms the state-of-the-art models. In contrast to this, our model has a complex structure in a deep network and takes longer training time. In future, we will extend this by incorporating the other graph model and deep learning framework to enhance the accuracy as well as reduce the training time.

1. References

[1] Jannach, D., Jugovac, M.: Measuring the business value of recommender systems. ACM Trans. Manage. Inf. Syst. 10(4), 1–23 (2019)

[2] Jannach, D., Ludewig, M., Lerche, L.: Session-based item recommendation in e-commerce: on short-term intents, reminders, trends and discounts. User Model. User-Adap. Inter. 27(3), 351–392 (2017)

[3] Jannach, D., Quadrana, M., Cremonesi, P.: Session-based recommendation. In: Ricci, F., Shapira, B., Rokach, L. (eds.) Recommender Systems Handbook. Springer, US (2021)

[4] Yuan, F.; Karatzoglou, A.; Arapakis, I.; Jose, J.M.; He, X. A simple convolution generative network forms the next item recommendation. In Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, Melbourne, VIC, Australia, 11–15 February 2019; pp. 582–590.

[5] Liu, Q.; Wu, S.; Wang, L.; and Tan, T. 2016. Predicting the next location: A recurrent model with spatial and temporal contexts. In AAAI, 194–200.

[6] Liu, Q.; Zeng, Y.; Mokhosi, R.; and Zhang, H. 2018. Stamp: Short-term attention/memory priority model for a session-based recommendation. In KDD, 1831–1839.

[7] Wang, S.; Cao, L.; Wang, Y.; Sheng, Q.Z.; Orgun, M.A.; Lian, D. A survey on session-based recommender systems. *ACM Comput. Surv.(CSUR)***2021**,*54*,1–38.

[8] Gupta, B.; Garg, D.FP-Tree Based Algorithms Analysis: FP Growth, COFI-Tree and CT-RO. *Int. J. Comput. Sci.Eng.***2011**,*3*,2691–2699.

[9] Song, W.; Yang, K. Personalized recommendation based on weighted sequence similarity. In *Practical Applications of Intelligent Systems*; Springer Berlin /Heidelberg, Germany,2014; pp. 657–666.

[10] Choi,K.;Yoo,D.;Kim,G.;Suh,Y.A hybrid online-product recommendation system: Combining implicit rating based collaborative filtering and sequential pattern analysis.*Electron. Commer.Res.Appl.***2012**,*11*,309–317.

[11] Liu, D.R.; Lai, C.H.; Lee, W.J.A hybrid of sequential rules and collaborative filtering for product Recommendation. *Inf.Sci.***2009**.

[12] Li, J.; Ren, P.; Chen, Z.; Ren, Z.; Lian, T.; Ma, J. Neural attentive session-based recommendation. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, Singapore, 6–10 November 2017; pp. 1419–1428.

[13] Fang, J. Session-based Recommendation with Self-Attention Networks. arXiv 2021, arXiv:2102.01922.

[14] Cho, K.; Merrienboer, V.B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, 25–29 October 2014; pp. 1724–173

[15] Hidasi, B.; Karatzoglou, A.; Baltrunas, L.; and Tikk, D. 2016a. Session-based recommendations with recurrent neural networks, In ICLR

[16] Wu, S.; Tang, Y.; Zhu, Y.; Wang, L.; Xie, X.; Tan, T. Session-based recommendation with graph neural networks. In Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019.

[17] Gupta, P.; Garg, D.; Malhotra, P.; Vig, L.; Shroff, G. NISER: Normalized item and session representations to handle popularity bias. In Proceedings of the 1st International Workshop on Graph Representation Learning on Its Applications (CIKM '19), Beijing, China, 3–7 November 2019.

[18] Chen, T.; Wong, R.C.W. Handling information loss of graph neural networks for session-based recommendation. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Virtual Event, 23–27 August 2020; pp. 1172–1180.

[19] Matheran, S.; Java, A.; Sahu, S.K.; Shaikh, A. Introducing Self-Attention to Target Attentive Graph Neural Networks. arXiv 2022, arXiv:2107.01516.

[20] Li, A.; Cheng, Z.; Liu, F.; Gao, Z.; Guan, W.; Peng, Y. Disentangled Graph Neural Networks for Session-based Recommendation. arXiv 2022, arXiv:2201.03482.

[21] Bach, N.X.; Long, D.H.; Phuong, T.M . Recurrent convolution a network for session-based Recommendations. *Neuro computing***2020**,*411*, 247–258.

[22] Zhang, J.; Ma, C.; Mu, X.; Zhao, P.; Zhong, C.; Ruhan, A . Recurrent convolution neural network for a session-based recommendation. *Neuro computing***2021**,*437*,157–167.

[23] Wang,N.;Wang,S.;Wang,Y.;Sheng, Q.Z.;Orgun ,M. Modelling local and global dependencies for next-item recommendations. At the International Conference on Web Information Systems Engineering; Springer: Cham, Switzerland, 2020; pp. 285–300.

[24] Chen,Z.;Silvestri,F.;Wang,J.;Zhang,Y.;Huang,Z.;Ahn,H.;Tolomei,G.Grease: Generate factual and counterfactual explanations of foreign-based recommendations.*arXiv***2022**, arXiv:2208.04222.

[25] Tan, J.; Xu, S.; Ge, Y.; Li, Y.; Chen, X.; Zhang, Y. Counter factual explainable recommendation. In Proceedings of the 30th ACM International Conference on Information & Knowledge Management, Virtual Event,1–5 November 2021; pp. 1784–1793.

[26] Wu, S.; Sun, F.; Zhang, W.; Xie, X.; Cui, B. Graph neural networks in recommender systems: A survey.ACM Comput. Surv. 2022, 55, 1–37.

[27] Gao, C.; Wang, X.; He, X.; Li, Y. Graph neural networks for recommender system. In Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, Tempe, AZ, USA, 21–25 February 2022; pp. 1623–1625.

[28] Wu, S.; Tang, Y.; Zhu, Y.; Wang, L.; Xie, X.; Tan, T. Session-based recommendation with graph neural networks. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 346–353.

[29] Xu, C.; Zhao, P.; Liu, Y.; Sheng, V.S.; Xu, J.; Zhuang, F.; Fang, J.; Zhou, X. Graph Contextualized Self-Attention Network for Session-based Recommendation. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, Macao, China, 10–16 August 2019; Volume 19, pp. 3940–3946.

[30] Yu, F.; Zhu, Y.; Liu, Q.; Wu, S.; Wang, L.; Tan, T. TAGNN: Target attentive graph neural networks for session-based recommendation. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, 25–30 July 2020; pp. 1921–1924.

[31] Pan, Z.; Cai, F.; Chen, W.; Chen, H.; De Rijke, M. Star graph neural networks for session-based recommendation. In Proceedings of the 29th ACM International Conference on Information &Knowledge

Management, Virtual Event, 19–23 October 2020; pp. 1195–1204.

[32] Wang, Z.; Wei, W.; Cong, G.; Li, X.L.; Mao, X.L.; Qiu, M. Global context enhanced graph neural networks for session-based recommendation. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, 25–30 July 2020; pp. 169–178.

[33] Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907. 2016 Sep 9.

[34] Yang, H., Kim, G., & Lee, J. H. (2022). Logit Averaging: Capturing Global Relation for Session-Based Recommendation. Applied Sciences, 12(9), 4256

[35] Li, J., Wang, Y., Song, G., & Zhang, N. (2023). A Session-Based Recommendation Model That Integrates the Temporal Sequence of Session Interactions and the Global Distance-Awareness of Items with Graph Neural Networks. *Applied Sciences*, *13*(24), 13031.

[36] Behera, Gopal, and Neeta Nain. "Trade-off between memory and model-based collaborative filtering recommender system." Proceedings of the International Conference on Paradigms of Communication, Computing and Data Sciences: PCCDS 2021. Springer Singapore, 2022.

.