

Article

Session-Enhanced Graph Neural Network Recommendation Model (SE-GNNRM)

Lifeng Yin, Pengyu Chen * and Guanghai Zheng 

School of Software, Dalian Jiaotong University, Dalian 116028, China; yinlifeng1030@163.com (L.Y.); zgh@djtu.edu.cn (G.Z.)

* Correspondence: cpywy@163.com

Abstract: Session-based recommendation aims to predict anonymous user actions. Many existing session recommendation models do not fully consider the impact of similar sessions on recommendation performance. Graph neural networks can better capture the conversion relationship of items within a session, but some intra-session conversion relationships are not conducive to recommendation, which requires model learning more representative session embeddings. To solve these problems, an improved session-enhanced graph neural network recommendation model, namely SE-GNNRM, is proposed in this paper. In our model, the complex transitions relationship of items and more representative item features are captured through graph neural network and self-attention mechanism in the encoding stage. Then, the attention mechanism is employed to combine short-term and long-term preferences to construct a global session graph and capture similar session information by using a graph attention network fused with similarity. In order to prove the effectiveness of the constructed SE-GNNRM model, three public data sets are selected here. The experiment results show that the SE-GNNRM outperforms the existing baseline models and is an effective model for session-based recommendation.

Keywords: session recommendation; graph neural networks; self-attention mechanism; transitions relationship



Citation: Yin, L.; Chen, P.; Zheng, G. Session-Enhanced Graph Neural Network Recommendation Model (SE-GNNRM). *Appl. Sci.* **2022**, *12*, 4314. <https://doi.org/10.3390/app12094314>

Academic Editor: Rocco Furferi

Received: 19 September 2021

Accepted: 22 April 2022

Published: 24 April 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the popularization of the Internet and the advent of the information age, the information that Internet users face and process every day is increasing exponentially. In order to select the content of interest, the recommendation system emerges as the times require. Most previous research work is based on uninterrupted user history for personalized recommendations, but in many services, such as YouTube and TikTok, due to limited storage resources, it is not possible or unnecessarily to track the user identity behavior. In other words, user profiles and long-term historical interactions are not available; only short-term session data from anonymous users are provided. For this scenario, research on session-based recommendation algorithms becomes more and more important [1]. The behavior information of user clicks generated within a certain time interval is a session sequence. In order to process this behavior information, the session-based recommendation system extracts the dependencies of multiple click items in the sequence and captures the importance of sequence prediction [2]. The interests of users in a certain period of time are the same and similar, which is important for the recommendation field, so the recommendation algorithms based on session sequence have received more attention in recent years [3].

At present, many session-based recommendation models have been born. In the early days, the session recommendation of Markov chains was mainly based on the user's previous behavior to predict the user's next behavior. With the development of deep learning, recurrent neural network (RNN) methods are gradually applied to session recommendation; such methods input user session click sequences into RNN to generate

recommendation results. RNNs are effective to some extent, but most existing RNN-based models have limited representational power in capturing complex transitions between items. Recently, recommendation algorithms based on graph neural networks (GNN) have also become popular. Wu et al. [4] adopted the GNN method to recommend users, converting session data into a graph structure and combining multiple layer gated recurrent units (GRU) to capture users' long-term interests and current preferences. This type of method inputs the session click item matrices constructed by the session graph and the session click sequence into the GNN for sequence modeling, which solves the problem of weak inter-session dependencies, and its recommendation effect is greatly improved, but it is not well filtered against useless dependencies. With the increasing popularity of deep neural networks, attention networks have been widely developed in many fields (e.g., computer vision, natural language processing, and recommendation systems), using self-attention mechanisms derived from the field of natural language processing to process session algorithms for sequences that are also emerging.

In order to overcome the above limitations, this paper combines graph neural networks (GNN for brevity), self-attention networks (SAN for brevity), soft attention mechanisms, and other related technologies to propose a session-enhanced graph neural network recommendation model (denoted as SE-GNNRM). This model includes an item embedding layer, a session embedding layer, a session interaction layer, and a prediction layer. The item embedding layer includes two modules, GNN and SAN. The item embedding obtained by GNN and SAN is combined through a joint function. In the session embedding layer, the output item embedding of the item embedding layer is combined with the location information, and then the session embedding is learned using attention. The session embedding at this time only represents a single session embedding, and similar sessions may have similar preferences. To solve this problem, this model designs a dynamic global session interaction layer for learning similar session embeddings. After obtaining the global session embedding, the candidate item scores are calculated through the prediction layer, and the final recommendation result is obtained.

The main work contributions of this paper are described as follows.

1. An improved session-enhanced graph neural network recommendation model based on a graph neural network and self-attention network, namely SE-GNNRM, is proposed to solve the problem of intra-session conversion relationships and representative session embeddings.
2. The graph neural network and self-attention mechanism are employed to capture the complex transitions relationship of items and more representative item features in the encoding stage.
3. The attention mechanism is employed to combine short-term and long-term preferences to construct a global session graph and capture similar session information by using a graph attention network fused with similarity.
4. For two evaluation metrics, extensive experiments on real-world data sets show that SE-GNNRM significantly outperforms existing baseline methods.

2. Related Work

2.1. Traditional Method

The early S-POP calculates the frequency of each item appearing in the session by counting the user's interactive behavior in the current session and recommends the top k items to the user. In the case that the current historical items are not full, the most popular items can be extracted from historical behavior as a supplement. S-POP considers the difference of each session but only considers the frequency of items appearing in the current session, which makes the session embedding learned by the model too simple, and it is difficult to fully represent the user's interest preference. Item-KNN [5] uses cosine similarity to define the degree of similarity between sessions and recommends items that are similar to previous clicks.

Shani et al. [6] consider the long-term effects and expected value Markov decision process; this method loses the balance between users' general preference and sequential behavior. Rendle et al. [7] proposed a hybrid method considering the combination of matrices factorization and collaborative filtering, embedding Markov chains in Euclidean space, calculating the transition probability between interactions according to Euclidean distance, and then inferring the user's next action based on the previous action. Because the method assumes that the current item only depends on one or a few adjacent items, the recommendation system based on the Markov chain focuses on the order conversion of two adjacent items, ignoring the interaction between multiple items. Markov chain-based methods are no longer suitable for today's recommendation scenarios.

2.2. Methods Based on Recurrent Neural Network (RNN)

With the rapid development of deep learning technology, many session recommendation algorithms based on deep learning have been born. The recurrent neural network (RNN) is a mainstream deep learning model for modeling sequential data [8]. Hidasi et al. [9] introduce RNN into the recommendation algorithm, regard the interaction of a session clicking a series of items as a sequence, and input the sequence into the model to predict the next clicked item. Li et al. [10] propose a neural attention recommendation model (NARM), which combines an attention mechanism with a multi-layer gated recurrent unit (GRU) encoder, which is a variant of RNN, in order to capture more representative session embeddings. The model uses the bilinear matching scheme based on the unified session embedding to calculate the recommendation score of each candidate item and then achieve better recommendation performance. The downside is that a session may contain multiple user selections and even noise, so they do not generate all the correct dependencies. Existing session-based recommendation methods mainly focus on the current session, ignoring collaboration information in historical sessions, which are sessions generated by other users and reflect similar user intentions to the current session. In response to this problem, Wang et al. [11] propose a collaborative session-based recommendation approach with parallel memory modules (CSRM), which uses a memory network to encode the current session, taking the information of the current session and neighbor sessions into account based on session suggestions to better predict the intent of the current session. One problem with session-based recommendations is how to generate accurate recommendations at the beginning of a session before knowing the user's current interests at the start of the session. Song et al. [12] transformed the short-session problem into a few-shot learning (FSL) problem, explicitly designing a local module based on GRU encoding and a global module considering similar historical sessions, and achieved state-of-the-art performance.

2.3. Methods Based on Graph Neural Network (GNN)

As an emerging artificial intelligence technology, graph learning (GL) has developed rapidly and has shown great promise in recent years. GL has a strong ability to learn relational data, and GL-based recommendation systems (RS) have been widely proposed and extensively studied. Most of the data in RS have a graph structure in nature, so GL has a positive impact on recommendation. Wu et al. [4] applied GNN to a sequence recommendation algorithm for the first time and proposed a GNN-based session recommendation model (SR-GNN). SR-GNN first models the session sequence information as a directed graph and uses gated GNN to capture the complex relationship and dependency transformation between each node in the session to obtain the node embedding vector. Then, SR-GNN uses the attention mechanism to obtain the global session embedding and finally combines the long- and short-term interest embedding to obtain the final embedding. To improve the embedding of session sequences, Xu et al. [13] propose an improved GNN-based graph contextual self-attention network model (GC-SAN). It exploits the complementarity between the self-attention network (SAN) and GNN to improve recommendation performance. Qiu et al. [14] propose a weighted graph attention network model as an item feature encoder, which helps transfer information between items more efficiently by learning to

assign different weights to different neighbors. Kim et al. [15] proposed a content-rich graph neural network with attention, which includes an attention mechanism for the rich content of each node when computing edge representations, capturing auxiliary behavioral information within the target domain. The above recommendation algorithms are mainly aimed at the modeling of the current session, ignoring the item conversion information between sessions, and the similarity and influence between different sessions are also very important for the recommendation results. Wang et al. [16] proposed a global context-enhanced graph neural network model (GCE-GNN) for session recommendation, which enhances the node's global embedding capability in the current session by sampling the neighbor nodes of nodes in other sessions.

2.4. Methods Based on Attention

Liu et al. [17] believe that it is meaningful to use RNN to capture general interest but do not consider the influence of current user interest on next behavior. In long sessions, RNN may experience a shift in interest due to user misclicks. Both short-term and long-term interests of users are important, but traditional RNN structures cannot distinguish and exploit both interests at the same time. Liu et al. propose a novel short-term attention memory priority model (STAMP), which is able to capture the user's general interests from the long-term memory of the session context while considering the user's current interests from the short-term memory of the last click.

In addition, with the rapid development of the natural language field, the emergence of Transformer completely eliminates the dependence of sequential processing on the architecture of recurrent connections, which is computationally inefficient and difficult to parallelize [18–23]. Transformer [24] only relies on the self-attention mechanism to capture the global dependencies between input and output. Transformer shows its powerful processing ability for text sequences and makes progress in sequence recommendation. Even in the case of session recommendation with shorter sequence length, Transformer outperforms the RNN recommendation method [25–31]. Sun et al. [32] believe that the unidirectional structure limits the ability of hidden embeddings in user behavior sequences and propose a sequential recommendation with bidirectional encoder embeddings from Transformer (BERT4Rec), which adopt deep bidirectional self-attention to model user behavior sequences, allowing each item in the user's historical behavior to fuse the information on the left and right sides for recommendation. Fang [33] proposed a session-based recommendation with the self-attention network model (SR-SAN), which captures the dependencies of all items in a session through a self-attention mechanism, and uses a single item vector to represent the current and global session embeddings. Yuan et al. [34] propose a dual sparse attention network model (DSAN), which uses a sparse transformation function to replace the original norm function in the self-attention of the embedding layer, which reduces the interference of the noise in the session on the item embedding and the session current, and global embeddings are learned using a target attention mechanism at the target embedding layer, which achieves suitable performance. The attention mechanism can shine in the field of recommendation because of the complex interactive behavior of its users when using various software platforms, and traditional deep learning techniques can well capture the ordering within sessions and the dependencies of item transitions [35]. However, it is difficult to identify complex interactive behaviors by relying too much on the order of training data, which limits the ability of recommendation systems to learn user dynamic preferences. The attention mechanism can identify interactions that are important for recommendation across distances.

3. Session-Enhanced Graph Neural Network Recommendation Model (SE-GNNRM)

The definition of the session recommendation model is given, and then the session-enhanced graph neural network recommendation model (SE-GNNRM) is elaborated. The detailed structure of the SE-GNNRM is shown in Figure 1. In Figure 1, v1–v5 represent the contained item in the current session.

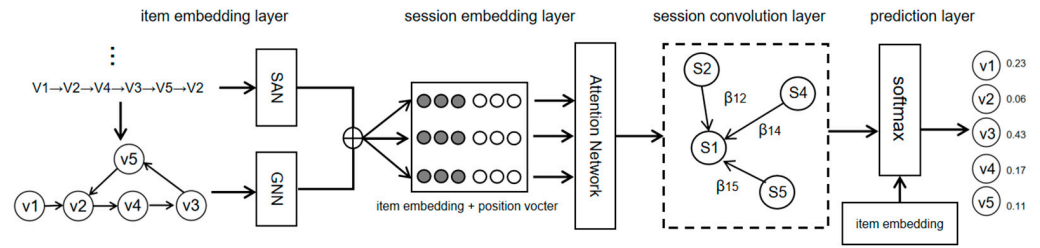


Figure 1. The detailed structure of SE-GNNRM.

3.1. Definition of Session Recommendation Model

Session-based recommendation aims to predict the item that the user clicks next after a given behavior sequence, only based on the user's current behavior information, without considering the user's historical clicks and user attributes, and other information. The problem is defined as follows: $V = [v_1, v_2, v_3, \dots, v_m]$ represents the set of all items, m is the total number of items. $S = [v_{s,1}, v_{s,2}, v_{s,3}, \dots, v_{s,n}]$ represents a sequence of sessions sorted by timestamp, where $v_{s,n} \in V$ is the item the user clicked on at step n . The next click predicted by session-based recommendations is $v_{s,n+1}$. Specifically, the recommendation model returns a list of predicted item scores $\hat{y} = [\hat{y}_1, \hat{y}_2, \hat{y}_3, \dots, \hat{y}_m]$, where $\hat{y}_i$ represents the predicted score associated with the item set v_i and recommends the top k candidate items with the highest score.

3.2. Detailed Structure of SE-GNNRM

SE-GNNRM includes an item embedding layer, a session embedding layer, a session interaction layer, and a prediction layer, which will be described in detail as follows.

3.2.1. Item Embedding Layer

The input to the model is the preprocessed sequence of sessions, and a directed graph of all session sequences is constructed. The item embedding layer contains two modules, a graph neural network (GNN) module and a self-attention (SAN). The directed graph of the session sequence is input into the GNN module, and then the item embedding is learned by using the powerful embedding learning ability of GNN for graph structure nodes. The session sequence is input into the SAN module, and the SAN learns the correlation between the session sequences through the self-attention mechanism in order to obtain a more representative embedding of the item.

GNN generates node embeddings on topological graphs to model the dependencies of item feature transformations, and GNN is suitable for session-based recommendations. Each session sequence S preprocessed is modeled as a directed graph $G_s = (V_s, E_s)$ as the input to the GNN module, where each node represents an item $v_{s,i} \in V_s$, and each connection in the directed graph represents connections between items in a session. Representing each item into a unified space, the resulting vector $v_i \in R^d$ is the d -dimensional embedding of v_i obtained using GNN. Based on the idea of RNN, Li et al. [36] introduce the GRU unit into GNN and propose a variant gated graph neural network (GGNN) to make it more suitable for processing sequence data. This model uses GGNN to extract node features. The updated propagation formula for node $v_{s,i}$ in session S is described as follows.

$$a_{s,i}^{(t)} = A_{s,i} [v_1^{(t-1)}, \dots, v_n^{(t-1)}]^T W_H + b_1 \quad (1)$$

$$z_{s,i}^{(t)} = \sigma(W_z a_{s,i}^{(t)} + U_z v_i^{(t-1)} + b_2) \quad (2)$$

$$r_{s,i}^{(t)} = \sigma(W_r a_{s,i}^{(t)} + U_r v_i^{(t-1)} + b_3) \quad (3)$$

$$\tilde{v}_i^{(t)} = \tanh(W_o a_{s,i}^{(t)} + U_o (r_{s,i}^{(t)} \odot v_i^{(t-1)})) \quad (4)$$

$$v_i^{(t)} = (1 - z_{s,i}^{(t)}) \odot v_i^{(t-1)} + z_{s,i}^{(t)} \odot \tilde{v}_i^{(t)} \quad (5)$$

where $W_H \in R^{d \times 2d}$, $W_z, U_z, W_r, U_r \in R^{d \times 3d}$, $W_o, U_o \in R^{d \times d}$ are the learnable weight matrices. $b_1 \in R^d$ and $b_2, b_3 \in R^{3d}$ are the bias vectors, and $z_{s,i}^{(t)}$ and $r_{s,i}^{(t)}$ are reset and update gates, respectively. The reset gate determines how to combine the new input information with the previous information. The update gate defines what the previous information is saved to the current time step. $[v_1^{(t-1)}, \dots, v_n^{(t-1)}]$ is the node vector embedding in session s , $\sigma(\cdot)$ is a sigmoid function, and $\odot$ is element-wise multiplication. $A_s \in R^{n \times 2n}$ is defined as the concatenation of two adjacent order matrices, $A_s^{(out)}$ and $A_s^{(in)}$, representing the weighted connection of incoming and outgoing edges of the session graph. The adjacency matrices A_s distinguish the types of edges in the directed graph and defines how the nodes in the directed graph are connected to each other through incoming and outgoing edges. $A_{s,i} \in R^{1 \times 2n}$ is the i -th row in A_s , which is the part corresponding to the node $v_{s,i}$. In addition, since duplicate items may appear in the session sequence, a normalized assignment is calculated for each edge, the initial value of the edge divided by the number of incoming or outgoing edges of the node.

Self-attention (SAN) is an attention mechanism that computes sequence embeddings related to different positions of a single sequence and has been successfully applied in many fields. The self-attention mechanism reduces the dependence on external information and is better at capturing the internal correlation of features. In natural language processing, it mainly solves the problem of long-distance dependence by calculating the mutual influence between words, filtering out important item information in the sequence, and assigning greater weight to it. In order to learn more representative session features, self-attention is used to encode session sequences in this paper.

The self-attention network in this paper follows the mechanism, which consists of an attention network and a feed-forward network. The session sequence $S = [v_{s,1}, v_{s,2}, v_{s,3}, \dots, v_{s,n}]$ is input into the attention network, and the attention function C consists of a map of the query and key-value pairs outputs:

$$C = softmax\left(\frac{(SW^Q)(SW^k)}{\sqrt{d}}\right)(SW^V) \quad (6)$$

where $W^Q, W^k, W^V \in R^{d \times d}$ is the learning weight matrices, and $\sqrt{d}$ is the normalization factor in preventing the calculation value of the inner product from being too large.

The feed-forward network consists of two layers of neural networks. *Relu* is selected as the first layer activation function to add non-linearity to the self-attention module. At the same time, transmission loss may occur in the self-attention network. Using residual connections can make full use of the bottom layer information. The feed-forward network layer formula is as follows:

$$F = H + Relu(CW_1 + b_4)W_2 + b_5 \quad (7)$$

where $W_1 \in R^{d \times 4d}$, $W_2 \in R^{4d \times d}$ is the weight matrices that can be learned, $b_4, b_5 \in R^d$ is the bias vector. For session S , $V_s = [v_{s,1}, v_{s,2}, v_{s,3}, \dots, v_{s,n}]$ and $F_s = [f_{s,1}, f_{s,2}, f_{s,3}, \dots, f_{s,n}]$ are obtained after encoding by the GNN module and the SAN module, respectively. Then, the two item embeddings are combined to complement each other. The joint function is described as follows.

$$O_s = V_s + \theta F_s \quad (8)$$

where θ is a weight parameter, which represents the importance of the item embedding learned by the SAN module, and is used to adjust the proportions of the two modules.

3.2.2. Session Embedding Layer

Although the item embedding layer of the model learns the transition dependencies of items in the session, it does not contain any recurrent or convolutional mechanisms, resulting in the model not fully considering the sequentiality of the session and therefore not knowing the position of the previous items in the session [37]. In session recommendation, each session is a sequence combination of a set of ordered items, where the position order encoding takes into account the different effects of relative positions between different elements in a sentence or a sequence [15,38]. In the model proposed in this paper, in order to strengthen the sequentiality of the learned session embedding, the reverse position information is incorporated into the session embedding through the position matrices $P = [p_1, p_2, p_3, \dots, p_n]$, where $p_i \in R^d$ is the position vector of position i . The session sequence $O = [o_{s,1}, o_{s,2}, o_{s,3}, \dots, o_{s,n}]$ is output by the joint function in the item embedding layer. After O is added to the position information, the i -th item is represented as:

$$z_i = \tanh(W_3[o_{s,i} \| p_{n-i+1}] + b_6) \quad (9)$$

where $W_3 \in R^{d \times 2d}$ is the weight matrices that can be learned, $b_6 \in R^d$ is the bias vector, $\tanh(\cdot)$ represents a bitangent activation function, $\|$ represents the splicing operation, and the item represents $v_{o,i}$ is concatenated with the position vector p_{n-i+1} , where n is the length of the current session. Inspired by STAMP and NARM, the session embedding that fuses the long-term and short-term interest preferences of a session can better reflect the characteristics of the current session. The purpose is to emphasize the importance of the current preference to adapt to the complex and changeable interaction behavior of users. Next, the soft attention mechanism is used to calculate the weighted summation of the attention coefficient α_i of the item in the session and the item embedding z_i to obtain the current session embedding h_c , which is described as follows.

$$h_c = \sum_{i=1}^n \alpha_i z_i \quad (10)$$

$$\alpha_i = q^T \sigma(W_4 z_i + W_5 z_n + W_6 z_{avg} + b_7) \quad (11)$$

$$z_{avg} = \frac{1}{n} \sum_{i=1}^n z_i \quad (12)$$

where $\sigma(\cdot)$ is a sigmoid function, where $W_4, W_5, W_6 \in R^{d \times d}$, $q \in R^d$ is the weight matrices used to calculate the attention coefficient α_i , and $b_7 \in R^d$ is the partial set vector. h_c is obtained by the weighted summation of the calculated attention coefficient α_i of the item in the session and the item embedding z_i . The target embedding z_n is defined as the last item of the input session sequence, z_{avg} is the average value represented by the items in the session, which represents a central feature of the session [39].

3.2.3. Session Interaction Layer

The target embedding and global embedding of the session actually only pay attention to the current session features but ignore the similar but different features between sessions, so this paper designs a session interaction layer. The GAT [40] aggregates neighbor node features through an attention mechanism, and the FGNN [14] uses GAT with weight information to learn node features. Inspired by the GAT and FGNN, this model calculates the co-occurrence degree of the same interactive items in existing sessions as the co-occurrence parameter ϕ_{ij} and integrates the graph attention network for feature propagation so that sessions with high similarity can learn from each other features that have an important impact on recommendation results. The final embedding of a session depends on the combination of the session itself and its neighborhood sessions.

In the session interaction layer, first, a global session graph is constructed, in which each node represents a session, and the co-occurrence parameter ϕ_{ij} of the items in the session is used to determine the connections of each node in the global session graph. Then, the current session embedding h_c output by the session embedding layer is used as

the initial input of the feature transfer function, and the similarity coefficient e_{ij} between nodes and neighboring nodes is obtained by calculating the inner product of nodes in the global session graph and their co-occurrence parameters. Finally, the softmax function is used to normalize e_{ij} to obtain the inter-session attention coefficient β_{ij} , which is described as follows.

$$e_{ij} = \text{LeakyRelu}([h_i \odot h_j \parallel \phi_{ij}] W_7) W_8 \quad (13)$$

$$\beta_{ij} = \frac{\exp(e_{ij})}{\sum_{h_k \in N_{h_i}} \exp(e_{ik})} \quad (14)$$

where, $\text{LeakyRelu}(\cdot)$ is a nonlinear function. Its negative slope is set to 0.2, $W_7 \in R^{(d+1) \times d}$, $W_8 \in R^d$ is a learnable weight matrix, $\odot$ is element-wise multiplication, N_{h_i} denotes the set of h_i neighborhood sessions. h_i , h_j , h_k are the outputs of the session embedding layer. The attention-aware neighborhood session embedding h_i^s is given as follows.

$$h_i^s = \sum_{h_k \in N_{h_i}} \beta_{ij} h_j \quad (15)$$

The final step of the session interaction layer will rely on the attention-aware neighborhood session embedding h_i^s to be aggregated with the current session embedding h_i . The neighborhood aggregation formula adopts a gating mechanism to autonomously select more meaningful session features, which is described as follows.

$$\gamma = \sigma(W_9 h_i + W_{10} h_i^s) \quad (16)$$

$$h_f = \gamma h_i + (1 - \gamma) h_i^s \quad (17)$$

where $W_9, W_{10} \in R^{d \times d}$ is a learnable weight matrix. γ is a gate obtained by a nonlinear activation function after calculating the linear transformation of h_i and h_i^s , h_f is the global embedding session.

3.2.4. Prediction Layer

In the prediction layer, the model evaluates the probability of the next clicked item in the session based on the session embedding h_f obtained by the session interaction layer, calculates the inner product of v_i and h_f as the recommendation score of each candidate item, and uses the softmax function to obtain the final recommendation probability.

$$\hat{y} = \text{softmax}(h_f^T \cdot v_i) \quad (18)$$

where v_i is the initial vector of the item, $\hat{y} \in R^m$ represents the probability that the candidate item is the next clicked item, and selecting the top k candidate items in $\hat{y}$ will be selected as the recommended item. To train the model, the loss function of the model is defined as the cross-entropy of the predicted value and the true value:

$$\mathcal{L}(\hat{y}) = - \sum_{i=1}^m y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) + \lambda \|\psi\|^2 \quad (19)$$

where y_i represents the one-hot encoded vector of the true value of the next click, ψ is the set of all learnable parameters, λ is the L_2 regularization parameter, and the model is trained using the back-propagation through time (BPTT) method.

4. Experiment and Analysis

In this section, the data set and experimental setup are first described, and then the designed experiments are presented to verify the performance of the proposed model SE-GNNRM, and the results under different experimental setups are analyzed in detail.

4.1. Data Sets

The real data sets used in the experiment are Yoochoose, Diginetica, and Tmall. The Yoochoose data sets are from RecSys Challenge 2015, which contains user click data of e-commerce websites within 6 months. This paper uses 1/64 of the Yoochoose data sets as the data sets to prevent the memory overflow caused by the Yoochoose data sets being too large. The Diginetica data sets are from CIKM Cup 2016, and this article only uses its transaction data. The Tmall data sets are from the IJCAI-15 competition data sets, which contain anonymous user shopping data on the Tmall shopping platform. In data preprocessing, all sessions of length 1 and items that appear less than five times in the data sets are filtered out, and the data sets are reordered in chronological order. For the Yoochoose data sets, the data from the last day are used as the test set and the remaining data as the training set, while for the Diginetica data sets, the data from the last seven days are used as the test set and the remaining data as the training set. For the Tmall data sets, the last 100 s are used as the test set, and the rest of the data are used as the training set. Then, the sequences and corresponding labels by splitting the input sequence are generated [41]. For session $S = [v_{s,1}, v_{s,2}, v_{s,3}, \dots, v_{s,k}]$, the sequences and corresponding labels are generated by sequence preprocessing, i.e., $([v_{s,1}], v_{s,2}), ([v_{s,1}, v_{s,2}], v_{s,3}), \dots, ([v_{s,1}, v_{s,2}, \dots, v_{s,k-1}], v_{s,k})$ is training and test sets for all three data sets. The statistics of the preprocessed data set are shown in Table 1.

Table 1. Statistics data for the data sets.

Data Set	Yoochoose1/64	Diginetica	Tmall
Clicks	557,248	982,961	818,479
Train	369,859	719,470	351,268
Test	55,898	60,858	25,898
Items	16,766	43,097	40,728
Average length	6.16	5.12	6.69

4.2. Baseline Algorithm

To verify the effectiveness of the proposed model, it is compared with the following methods. The baseline method is described as follows.

1. Item-KNN [5] uses cosine similarity to define the degree of similarity between sessions and recommends items that are similar to previous clicks.
2. FPMC [7] is a sequence recommendation method based on Markov chain and matrices factorization.
3. GRU4REC [9] uses RNN to model user sequences for session-based proposals.
4. NARM [10] uses an RNN with an attention mechanism to capture the main purpose and sequential behavior of users, adding an attention mechanism to capture the user's general interest and current interest in the current session.
5. CSRM [11] consists of an internal and an external memory encoder to model the preferences of the current session and neighborhood sessions in order to better predict the intent of the current session.
6. SR-GNN [4] uses GNN to capture complex item information transformation and uses an attention mechanism to capture long-term and short-term interests.
7. FGNN [14] learns item vectors through a weighted attention layer and learns session features through a session graph feature extraction layer.
8. STAMP [17] uses an attention network to capture the user's general interest and current interest in the current session.
9. CoSAN [42] uses a self-attention network to learn the current session embedding and collaborative feature information of items from neighborhood sessions.

4.3. Parameter Settings

The experimental parameters in this paper are set as follows, the dimension of item embedding and the training batch size are set to 100. Co-occurrence parameter ϕ_{ij} of session interaction layer selection range is [0.0–0.9]. Use random initialization for each vector, with all parameters having a mean of 0 and a standard deviation of 0.1. All parameters are optimized using the cross-entropy loss function and the mini-batch Adam algorithm, where the initial learning rate is set to 0.001, the number of epochs is set to 30, and the decay will be 0.1 after every 3 epochs, using L2 regularization to prevent the algorithm from overfitting, L2 regularization coefficient selection range is [1×10^{-3} , 5×10^{-4} , 1×10^{-4} , 5×10^{-5} , 1×10^{-5}]. For a fair comparison, the dimension of the item embedding of the baseline algorithm is set to 100, and other basic hyperparameters follow what is shown in the baseline algorithm literature. The number of hidden units in both GRU4REC and NARM is set to 100, the memory of CSRMM is set to 100, the number of attention layers is 3 in FGNN, and the multi-head mechanism used is set to 8. The number of CoSAN neighbor sessions is set to 5.

4.4. Evaluation Metrics

Two metrics were selected for the experiments to evaluate the compared methods. P@20 (precision) is widely used as a measure of prediction accuracy, which represents the proportion of correctly recommended items among the top 20 items. MRR@20 (mean reciprocal rank) is the average of the reciprocal rank of correctly recommended items, where a larger MRR value represents a correct recommendation at the top of the ranking list.

In order to verify the performance of the model, two groups of experiments are designed in this paper, and the experimental results are analyzed. The first set of comparative experiments compares the performance of SE-GNNRM with nine baseline models. The results of the comparative experiments are shown in Table 2. The second set is hyperparameter experiments to investigate the effect of the co-occurrence parameter ϕ_{ij} on model performance.

Table 2. Comparative experimental results.

Models (%)	Yoochoose1/64		Diginetica		Tmall	
	P@20	MRR@20	P@20	MRR@20	P@20	MRR@20
FPMC	45.62	15.01	26.53	6.95	16.06	7.32
Item-KNN	51.60	21.81	35.75	11.57	9.15	3.31
GRU4REC	60.64	22.89	30.79	15.46	10.83	5.89
NARM	68.32	28.63	48.43	16.11	23.30	10.70
CSRMM	71.45	30.36	50.55	16.38	29.46	13.96
SR-GNN	70.57	30.94	50.73	17.59	27.57	13.72
FGNN	71.12	31.68	51.36	18.47	25.24	10.39
STAMP	68.74	28.67	46.62	15.13	26.47	13.36
CoSAN	69.71	29.92	48.34	15.22	27.21	13.12
SE-GNNRM	72.76	32.18	52.84	18.49	32.70	15.61

4.5. Results and Analysis

4.5.1. Comparative Experiment

The experiment uses TensorFlow and PyTorch frameworks and uses Python language to simulate the existing nine baseline models and SE-GNNRM and compares and analyzes their performance. The comparison results are shown in Table 2.

From Table 2, it can be found that the SE-GNNRM proposed in this paper achieves the best performance on the two evaluation metrics P@20 and MRR@20. On the Yoochoose1/64 data sets, it is 1.42% and 1.58% higher than the suboptimal algorithm, on the Diginetica data set, it is 1.53% and 0.1% higher than the second-best algorithm, and on the Tmall

data set, it is 10.9% and 11.8% higher than the second-best algorithm. The SE-GNNRM effectively demonstrates its ability to recommend all three data sets.

In the early recommendation models, the FPMC model limits dependencies in the session sequence, so the recommendation effect is poor. Item-KNN outperforms Markov chain-based algorithms, but Item-KNN only recommends highly similar items and does not fully consider the problem of order transitions in sessions.

Compared with traditional algorithms, deep learning-based models explicitly model the user's global behavioral preferences and take into account the transition between the user's previous behavior and the next click, effectively proving their ability to capture session sequences. There is a stronger ability to sequentially transform relationships. GRU4REC, a classic work on modeling session sequences using RNNs, proves that encoding the current session with GRU units is more suitable for session recommendation problems than traditional RNN and LSTM on session recommendation problems and takes the last layer as a general preference for sessions. However, the interest preferences represented by session embeddings are variable, and GRU4REC does not distinguish current preferences from general preferences. On the basis of GRU4REC, NARM solves this problem well. After using the GRU unit to encode the current session to obtain the item embedding, it dynamically selects and linearly combines different parts of the input session sequence through the soft attention mechanism. The experimental results also demonstrate the effectiveness of this mechanism for learning session preferences. CSRM also uses a GRU-based RNN to model the session sequence while also considering the current and general preferences of the session. Furthermore, CSRM designs an external memory encoding layer for learning the collaborative information of adjacent sessions, which makes it have better recommendation performance than many models modeling a single session.

In the comparative experimental results, the GNN-based algorithm is more competitive than the RNN-based algorithm. SR-GNN further considers the dependencies of item transitions in a session by modeling each session as a directed graph. The items in the session are the nodes in the graph, which can capture more complex and implicit dependencies within the session. Experimental results show that even though SR-GNN is based on single session modeling, its performance is almost on par with algorithm CSRM, which considers multiple sessions, proving that it is meaningful to use GNN to model session sequences. FGNN designs a weight map attention layer for learning the item embedding in the session graph and also uses a GRU-based readout function to abstract the session embedding. FGNN uses a variety of deep learning techniques and achieves pretty suitable performance in Yoocoose1/64 and Diginetica.

STAMP is an algorithm that uses an attention mechanism to explicitly represent the last item of the session and the average features of the session and items embedding in the session to capture the preference of the session. The recent session recommendation algorithm based on self-attention has shined in many fields. CoSAN uses the strong encoding ability of self-attention for sequence data to learn session sequences from the perspective of similarity and location information between items, incorporates the collaborative information based on the adjacent session identification, and also gets quite suitable results. The experimental results show that the algorithm of the attention mechanism outperforms some existing deep learning algorithms and is a powerful choice for dealing with the session recommendation problem.

In contrast, the SE-GNNRM algorithm proposed in this paper uses both a GNN and a self-attention mechanism to encode session sequences and designs a fusion function to fuse the two session vectors to complement each other. Then, in order to reasonably learn the long-term and short-term preferences of the session, the soft attention mechanism fused with reverse position information is used to obtain the session embedding, which not only takes into account the transformation of item information in the session but also strengthens the influence of the sequential position of the items. Finally, a session interaction layer is designed based on the idea that session preferences with high similarity may also be similar, which introduces the influence of multi-session collaborative information into the algorithm.

The preference information of neighbor sessions is further aggregated through the graph attention network on the constructed global session graph, which makes it more fully and efficiently process session sequence information. In summary, the experimental results and analysis demonstrate the rationality and effectiveness of the SE-GNNRM proposed in this paper.

4.5.2. Co-Occurrence Parameter Analysis

This subsection shows the effect of choosing different co-occurrence parameters ϕ_{ij} between 0.0 and 0.9 for the three data sets on the recommendation performance. In the session interaction layer, a dynamic global session graph needs to be constructed, and the connections between nodes in the global session graph are determined according to the co-occurrence parameters. There are edges only between nodes that meet the co-occurrence parameter threshold so that the neighbor sessions of the current session can be confirmed according to the global session graph. Therefore, for the session interaction layer, how to choose the appropriate co-occurrence parameters needs to be verified and discussed through experiments. The following figures show the performance of SEGR when selecting different co-occurrence parameters for three data sets. It can be observed from Figure 2 that the trend is flat between 0.3 and 0.8 on the Tmall data set, and SEGR finally chooses 0.6 as the co-occurrence parameter for the Tmall data sets. On the Yoochoose1/64 and Diginetica data sets, the best results are achieved at 0.5 and 0.4, respectively. It can be concluded that the optimal values on the three data sets tend to be in the middle range of the interval. This is explainable. If the selection of neighbor sessions is too low and noise may be introduced to affect the performance of the recommendation algorithm, and if the selection is too high, similar neighbor features cannot be fully learned, or even there are no neighbors.

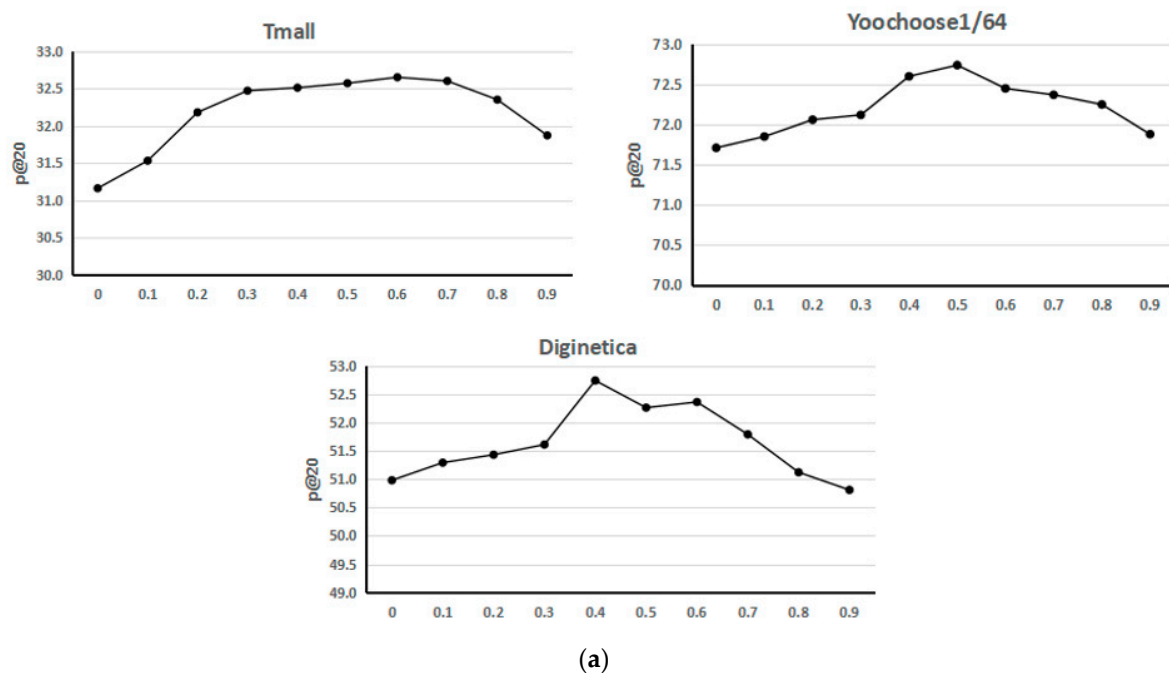


Figure 2. Cont.

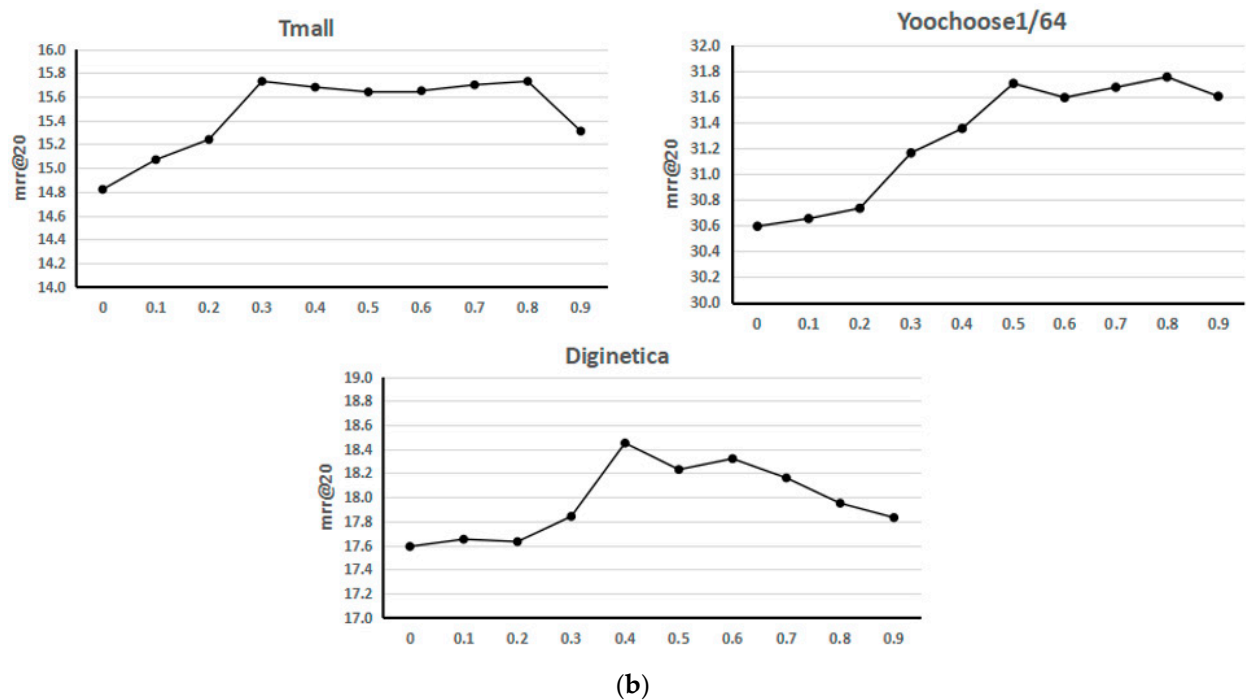


Figure 2. The effect of co-occurrence parameter ϕ_{ij} in three. (a) The effect of co-occurrence parameter ϕ_{ij} on P@20. (b) The effect of co-occurrence parameter ϕ_{ij} on MRR@20.

5. Conclusions

For many existing session recommendation models, more consideration is given to the order transition within the session, ignoring the impact of similar sessions on the recommendation performance. Some intra-session transition relationships captured by the GNN are not conducive to recommendation. In this paper, an improved session recommendation model based on GNN and attention networks, namely a session-enhanced graph neural network (SE-GNNRM), is proposed. Firstly, the model is modeled based on the session sequence, and the initial session sequence is encoded by GNN and SAN, respectively, and the obtained item vector takes into account the complex item transition and the importance of the item at the same time. Secondly, in order to enable the model to learn similar session information, a session interaction layer is designed. A global session graph is constructed, and the features between similar sessions are learned through an attention mechanism that fuses the similarity information between sessions. Extensive experiments on three real data sets demonstrate the effectiveness of the model. In the future, it is hoped that the recommendation performance of the model can be improved through more efficient graph representation learning and the introduction of more meaningful auxiliary information.

Author Contributions: Methodology, P.C.; software, L.Y.; validation, L.Y., P.C. and G.Z.; formal analysis, G.Z.; writing—original draft preparation, L.Y.; writing—review and editing, G.Z.; visualization, P.C.; funding acquisition, L.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China grant number [61771087].

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Gao, C.; Zheng, Y.; Li, N.; Li, Y.; Qin, Y.; Piao, J.; Quan, Y.; Chang, J.; Jin, D.; He, X.; et al. Graph neural networks for recommender systems: Challenges, methods, and directions. *arXiv* **2021**, arXiv:2109.12843.
2. 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. [\[CrossRef\]](#)
3. Latifi, S.; Mauro, N.; Jannach, D. Session-aware recommendation: A surprising quest for the state-of-the-art. *Inf. Sci.* **2021**, *573*, 291–315. [\[CrossRef\]](#)
4. Wu, S.; Tang, Y.; Zhu, Y.; Wang, L.; Xie, X.; Tan, T. Session-based Recommendation with Graph Neural Network. In Proceedings of the National Conference on Artificial Intelligence, Hilton Hawaiian Village, Honolulu, HI, USA, 27 January–1 February 2019; pp. 346–353.
5. Sarwar, B.; Karypis, G.; Konstan, J.; Riedl, J. Item-based collaborative filtering recommendation algorithms. In Proceedings of the 10th International Conference on World Wide Web, Hong Kong, China, 1–5 May 2001; pp. 285–295.
6. Shani, G.; Heckerman, D.; Brafman, R.I.; Boutilier, C. An MDP-based recommender system. *J. Mach. Learn. Res.* **2005**, *6*, 1265–1295.
7. Rendle, S.; Freudenthaler, C.; Schmidt-Thieme, L. Factorizing personalized markov chains for next-basket recommendation. In Proceedings of the 19th International Conference on World Wide Web, Raleigh, NC, USA, 26–30 April 2010; pp. 811–820.
8. Wang, S.; Hu, L.; Wang, Y.; He, X.; Sheng, Q.Z.; Orgun, M.; Cao, L.; Wang, N.; Ricci, F.; Yu, P.S. Graph Learning Approaches to Recommender Systems: A Review. *arXiv* **2020**, arXiv:2004.11718.
9. Hidasi, B.; Quadrana, M.; Karatzoglou, A.; Tikk, D. Parallel recurrent neural network architectures for feature-rich session-based recommendations. In Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, 15–19 September 2016; pp. 241–248.
10. Li, J.; Ren, P.; Chen, Z.; Ren, Z.; Lian, T.; Ma, J. Neural attentive session-based recommendation. In Proceedings of the Conference on Information and Knowledge Management, Singapore, 6–10 November 2017; pp. 1419–1428.
11. Wang, M.; Ren, P.; Mei, L.; Chen, Z.; Ma, J.; de Rijke, M. A collaborative session-based recommendation approach with parallel memory modules. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Paris, France, 21–25 July 2019; pp. 345–354.
12. Song, W.; Wang, S.; Wang, Y.; Wang, S. Next-item Recommendations in Short Sessions. In Proceedings of the RecSys’21: Fifteenth ACM Conference on Recommender Systems, Amsterdam, The Netherlands, 27 September–1 October 2021; pp. 282–291.
13. 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 International Joint Conference on Artificial Intelligence, Macao, China, 10–16 August 2019; pp. 3940–3946.
14. Qiu, R.; Li, J.; Huang, Z.; Yin, H. Rethinking the Item Order in Session-based Recommendation with Graph Neural Networks. In Proceedings of the 28th International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019; pp. 579–588.
15. Kim, J.; Lamb, A.; Woodhead, S.; Jones, S.P.; Zheng, C.; Allamanis, M. CoRGi: Content-Rich Graph Neural Networks with Attention. *arXiv* **2021**, arXiv:2110.04866.
16. 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 SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, 25–30 July 2020; pp. 169–178.
17. Liu, Q.; Zeng, Y.; Mokhosi, R.; Zhang, H. STAMP: Short-term attention/memory priority model for session-based recommendation. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018; pp. 1831–1839.
18. Li, Z.G.; Chen, H.; Ma, X.C.; Chen, H.Y.; Ma, Z. Triple Pseudo-Siamese network with hybrid attention mechanism for welding defect detection. *Mater. Des.* **2022**, *217*, 110645. [\[CrossRef\]](#)
19. Deng, W.; Li, Z.; Li, X.; Chen, H.; Zhao, H. Compound fault diagnosis using optimized MCKD and sparse representation for rolling bearings. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 1–9. [\[CrossRef\]](#)
20. Cui, H.; Guan, Y.; Chen, H. Rolling element fault diagnosis based on VMD and sensitivity MCKD. *IEEE Access* **2021**, *9*, 120297–120308. [\[CrossRef\]](#)
21. Zhang, Z.H.; Min, F.; Chen, G.S.; Shen, S.P.; Wen, Z.C.; Zhou, X.B. Tri-partition state alphabet-based sequential pattern for multivariate time series. *Cogn. Comput.* **2021**, 1–19. [\[CrossRef\]](#)
22. Chen, H.; Zhang, Q.; Luo, J. An enhanced Bacterial Foraging Optimization and its application for training kernel extreme learning machine. *Appl. Soft Comput.* **2020**, *86*, 105884. [\[CrossRef\]](#)
23. Ran, X.; Zhou, X.; Lei, M.; Tepsan, W.; Deng, W. A novel k-means clustering algorithm with a noise algorithm for capturing urban hotspots. *Appl. Sci.* **2021**, *11*, 11202. [\[CrossRef\]](#)
24. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5998–6008.
25. Shao, H.D.; Lin, J.; Zhang, L.W.; Galar, D.; Kumar, U. A novel approach of multisensory fusion to collaborative fault diagnosis in maintenance. *Inf. Fusion* **2021**, *74*, 65–76. [\[CrossRef\]](#)
26. Li, T.Y.; Shi, J.Y.; Deng, W.; Hu, Z.D. Pyramid particle swarm optimization with novel strategies of competition and cooperation. *Appl. Soft Comput.* **2022**, *121*, 108731. [\[CrossRef\]](#)

27. Li, G.; Li, Y.; Chen, H.; Deng, W. Fractional-Order Controller for Course-Keeping of Underactuated Surface Vessels Based on Frequency Domain Specification and Improved Particle Swarm Optimization Algorithm. *Appl. Sci.* **2022**, *12*, 3139. [[CrossRef](#)]
28. He, Z.Y.; Shao, H.D.; Zhong, X.; Zhao, X.Z. Ensemble transfer CNNs driven by multi-channel signals for fault diagnosis of rotating machinery cross working conditions. *Knowl.-Based Syst.* **2020**, *207*, 106396. [[CrossRef](#)]
29. Wei, Y.Y.; Zhou, Y.Q.; Luo, Q.F.; Deng, W. Optimal reactive power dispatch using an improved slime mould algorithm. *Energy Rep.* **2021**, *7*, 8742–8759. [[CrossRef](#)]
30. Deng, W.; Zhang, X.X.; Zhou, Y.Q.; Liu, Y.; Zhou, X.B.; Chen, H.L.; Zhao, H.M. An enhanced fast non-dominated solution sorting genetic algorithm for multi-objective problems. *Inf. Sci.* **2022**, *585*, 441–453. [[CrossRef](#)]
31. Zhang, X.; Wang, H.; Du, C.; Fan, X.; Cui, L.; Chen, H.; Deng, F.; Tong, Q.; He, M.; Yang, M.; et al. Custom-molded offloading footwear effectively prevents recurrence and amputation, and lowers mortality rates in high-risk diabetic foot patients: A multicenter, prospective observational study. *Diabetes Metab. Syndr. Obes. Targets Ther.* **2022**, *15*, 103–109. [[CrossRef](#)]
32. Sun, F.; Liu, J.; Wu, J.; Pei, C.; Lin, X.; Ou, W.; Jiang, P. BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019; pp. 1441–1450.
33. Fang, J. Session-based Recommendation with Self-Attention Networks. *arXiv* **2021**, arXiv:2102.01922.
34. Yuan, J.; Song, Z.; Sun, M.; Wang, X.; Zhao, W.X. Dual Sparse Attention Network for Session-based Recommendation. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual Event, 2–9 February 2021; pp. 4635–4643.
35. Zhang, S.; Yao, L.; Sun, A.; Tay, Y. Deep Learning Based Recommender System: A Survey and New Perspectives. *ACM Comput. Surv.* **2019**, *52*, 1–38. [[CrossRef](#)]
36. Li, Y.; Tarlow, D.; Brockschmidt, M.; Zemel, R. Gated graph sequence neural networks. In Proceedings of the 4th International Conference on Learning Representations, San Juan, Puerto Rico, 2–4 May 2016.
37. Kang, W.C.; McAuley, J. Self-Attentive Sequential Recommendation. In Proceedings of the IEEE International Conference on Data Mining, Singapore, 17–20 November 2018; pp. 197–206.
38. Xia, X.; Yin, H.; Yu, J.; Wang, Q.; Cui, L.; Zhang, X. Self-Supervised Hypergraph Convolutional Networks for Session-based Recommendation. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual Event, 2–9 February 2021; pp. 4503–4511.
39. Duan, H.; Zhu, J. Context-aware short-term interest first model for session-based recommendation. *arXiv* **2021**, arXiv:2103.15514.
40. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; Bengio, Y. Graph attention networks. *arXiv* **2017**, arXiv:1710.10903.
41. Tan, Y.K.; Xu, X.; Liu, Y. Improved recurrent neural networks for session-based recommendations. In Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, Boston, MA, USA, 15 September 2016; pp. 17–22.
42. Luo, A.; Zhao, P.; Liu, Y.; Zhuang, F.; Wang, D.; Xu, J.; Fang, J.; Sheng, V.S. Collaborative Self-Attention Network for Session-based Recommendation. In Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, Yokohama, Japan, 11–17 July 2020; pp. 2591–2597.