



Research Article

Toward Sequential Recommendation Model for Long-Term Interest Memory and Nearest Neighbor Influence

Hongyun Cai ^{1,2}, Jie Meng ^{1,2}, Jichao Ren,^{1,2} and Shilin Yuan^{1,2}

¹School of Cyber Security and Computer, Hebei University, Baoding, 071000 Hebei, China

²Key Laboratory on High Trusted Information System in Hebei Province, Hebei University, Baoding, 071000 Hebei, China

Correspondence should be addressed to Jie Meng; mengjie_hbu@163.com

Received 10 May 2022; Accepted 10 September 2022; Published 27 September 2022

Academic Editor: Changyan Yi

Copyright © 2022 Hongyun Cai et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Sequential recommendation can make predictions by fitting users' changing interests based on the users' continuous historical behavior sequences. Currently, many existing sequential recommendation methods put more emphasis upon users' recent preference (i.e., short-term interests), but simplify or even ignore the influence of users' long-term interests, resulting in important interest features of users not being effectively mined. Moreover, users' real intentions may not be fully captured by only focusing on their behavior histories, because users' interests are diverse and dynamic. To solve the above problems, we propose a novel sequential recommendation model for long-term interest memory and nearest neighbor influence. Firstly, item embeddings based on item similarity and dependency are constructed to alleviate the problem of data sparsity in users' recent interest history. Secondly, in order to effectively capture long-term interests, the long sequence is divided into multiple nonoverlapping subsequences. For these subsequences, the graph attention network with node importance factor is designed to fully extract the main interests of subsequences, and LSTM is introduced to learn the dynamic changes of interest among subsequences. Long-term interests of users are modeled through complex structure within subsequences and sequential dependencies among subsequences. Finally, the user's neighbor representation is introduced, and a gating module is designed to integrate the user's neighbor information and self-interests. The influence of users' short-term and long-term interests on prediction is dynamically controlled by considering nearby features in the gating network. The experimental results on two public datasets show that the proposed sequential recommendation model can outperform the baseline methods in hit rate (HR@K) and normalized discounted cumulative gain (NDCG@K).

1. Introduction

With the rapid development of information technology, complex and diverse data are flooding people's lives. To deal with the problem of information overload, recommender systems have emerged as a pervasive part of online platforms. Different types of recommendation model have been developed, e.g., collaborative filtering recommendation [1], sequential recommendation [2], social recommendation [3], and group recommendation [4]. Among these models, sequential recommendation can effectively learn the changing of user's interests and provide more accurate recommendations, which has become a research hotspot in recent years [5, 6].

Nowadays, deep learning models (e.g., convolutional neural networks (CNN) [7], recurrent neural networks

(RNN) [8], attention mechanisms [9], and graph neural networks (GNN) [10]) are widely used in sequential recommender systems. However, the existing sequential recommendation models based on deep learning mainly focus on the users' behavioral interactions in the recent period and use the users' short-term interests to predict their subsequent choices, while the rich feature information contained in the users' long-term historical behaviors has not been further explored. In fact, people usually have stable and dynamically changing interests though user interests are complex and diverse. Previous studies [11–14] also showed that the selection of users in recommender systems is not only affected by their recent intentions but also related to their long-term stable interests. However, due to the long length of user behavior sequences and the complex

relationship between items, it is difficult to effectively learn users' long-term interests. Therefore, in sequential recommendation, the recommendation performance can be improved if we can further excavate the stable features of users' long-term interests on the basis of the dynamic changes of short-term interests.

In addition, the gating network can adaptively control the degree of information retention; the short-term interests and long-term interests of users can be dynamically fused through the gating network in sequential recommendation [15, 16]. However, if the prediction is only based on the user's own historical behaviors, the attention of the model is limited to the interest memory in the user's historical behaviors, which affects the recommendation effect. In fact, users are also interested in items selected by their similar neighbors [17]. The existing recommendation methods considering the nearest neighbor influence [18, 19] lack the dual attention to user behavior sequence and neighbor users or adopt a simple fusion approach [20] ignoring the interaction between the two aspects.

To address the above problems, we propose a sequential recommendation model for long-term interest memory and nearest neighbor influence (SRLIN for short). The proposed model deeply mines the user's long-term interests on the basis of learning the user's recent interests and incorporates the user's neighbor influence into the gating network. Specifically, based on the two different perspectives of item similarity and dependency, the item embeddings are first generated. Interest changes within recent sequences are learned by using a bidirectional LSTM (i.e., BiLSTM), and then, the self-attention network is used to obtain the user's short-term interests. Secondly, to effectively capture long-term interests, we propose a long-term interest modeling method including the interest extraction layer and the interest fusion layer. For each user, its long sequence is divided into multiple disjoint subsequences. In the interest extraction layer, the graph attention network with node importance factor (NIF_GAT for short) is designed, which can fully extract the main interest features of subsequences by learning the importance of different items in each subsequence and the complex relationship between items. In the interest fusion layer, we use the LSTM to learn the sequential dependencies of interest features in different time periods. The user's long-term interest representation is obtained through this hierarchical structure. Finally, the neighbor features of each user are extracted based on the ordered user sequences of the items, and the gating network that considers the neighbor features is introduced to adjust the influence of short-term interest representation, long-term interest representation, and nearest neighbor representation on prediction.

The main contributions of this paper are summarized as follows:

- (1) To effectively alleviate the sparsity problem of sequence recommendation, we propose an item embedding method based on item similarity and dependency
- (2) To more accurately capture user stable and changing long-term interests, we propose a long-term interest

modeling method including the interest extraction layer and the interest fusion layer. In the interest extraction layer, we first model the complex structure of subsequences and learn the main interest features within different subsequences by the improved graph attention network with node importance factor. Then, we use LSTM to learn the sequential dependencies of interests among different subsequences in the interest fusion layer

- (3) To further improve the recommendation performance, we design a gating fusion module based on the influence of neighbors, which can automatically adjust the weights of short-term and long-term interests by considering the neighbor information and deal with the situation where it is difficult to fully capture the user's intention only by relying on the user's own interests
- (4) Experimental results on two public datasets, i.e., MovieLens 1M and JD, show that our SRLIN model can outperform the state-of-the-art sequential recommendation methods

2. Related Work

2.1. General Sequential Recommendation. Ding et al. [21] and Lathia et al. [22] studied recommendation models based on time awareness. Based on the collaborative filtering algorithm, a time decay factor was introduced to describe the change of user interests over time. Subsequently, Rendle et al. [23] proposed an FPMC model based on matrix factorization and Markov chains, which combined the sequential behaviors of different users by establishing a three-dimensional transformation matrix, and used a first-order Markov model to model the user's historical behaviors. The FPMC model fully integrates the advantages of matrix factorization and Markov chains and improves the accuracy of sequential recommendation method. He et al. [24] extended on FPMC, which adopted a higher-order Markov chain to learn the complex relationship of data [25]. In addition, Sahoo et al. [26] proposed a collaborative filtering recommendation method based on hidden Markov model. Considering that the traditional Markov chain is difficult to model long-term historical sequences of users, Lonjarret et al. [27] proposed the REBUS model which uses frequent sequences to capture the most relevant parts of user history for recommendations.

2.2. Deep Learning-Based Sequential Recommendation. Deep learning can automatically learn features, which has attracted extensive attention in sequence recommendation in recent years. Tang et al. [7] transformed sequence data into "images" with temporal information and used convolutional filters to learn sequence features. Kang et al. [9] adopted a stacked self-attention mechanism to effectively capture the high-order features of the sequence, in which the model structure is similar to the Encoder of the Transformer. Hidasi et al. [8] proposed a new loss function used in the recurrent neural network model, which can alleviate

the gradient vanishing problem of sequential recommendation. Among many deep learning models, recurrent neural networks have received extensive attention due to their unique properties of sequential learning. The recurrent neural network uses the output of the previous node as a new part of the input and does not add additional biases, which can track the user's interest changes in essence. However, these methods only take random initialization of item numbers as input, which cannot clearly describe the relationship between items and have poor interpretability. Therefore, Huang et al. [28] proposed the ATST-LSTM model for the next POI recommendation, which applies the time interval and distance interval as auxiliary information on the time steps of the LSTM. The application of auxiliary information can greatly alleviate the problem of data sparsity and improve the prediction effect. However, this kind of auxiliary information only relies on the user's own historical behaviors, which cannot help to fully capture the user's implicit interests. Therefore, we measure the similarity and dependency between items from a global perspective; based on which, item embeddings can be generated.

In addition, the above models mainly focus on users' recent behaviors. However, some studies have shown that in addition to the recent interactions, the user's interests are also affected by her/his early choices [2]. Therefore, some scholars divided user history records into recent sequences and global sequences and proposed long-short term interests fusion models. Gan et al. [29] proposed an R-RNN model, which uses LSTM to focus on user's recent behaviors and applies MLP to fuse long-term and short-term interests. Ying et al. [30] proposed a hierarchical attention network, which uses the attention mechanism to learn short-term interests and fuse long-term with short-term interests. The fusion of long-term and short-term interests comprehensively considers long-term and short-term features, which can improve the accuracy of recommendation. But the above methods just adopt some simple ways to learn the users' long-term interests. To make better use of the rich information contained in long sequences and improve the problem of imperfect long-term interest modeling, Lv et al. [15] used an attention mechanism to learn different aspects of long-term interests and introduced a gating module to extract features related to short-term interests in long-term interests. Lin et al. [31] improved the attention mechanism in long-term interest learning, which improved the recommendation performance. However, it is difficult to track the dynamic change trend of user interests by directly modeling the whole long sequence, which is prone to the phenomenon of recommendation performance degradation. Quadrana et al. [32] proposed a hierarchical RNN model HRNN. For long sequences, the model implements RNN-based session modeling at the bottom layer for each session and uses RNN at a higher level to track the evolution of user interests for cross-session learning. The splitting of long sequences can reduce the difficulty of overall modeling and simplify complex problems. The experimental results of HRNN model also prove that the hierarchical model can obtain better recommendation performance than the overall modeling of long sequences. However, the model is susceptible to

noise. The reason is that the underlying RNN of HRNN model performs the strict order in the session. For example, when a user is browsing the shopping page, he or she may click some products out of curiosity. The interest offset caused by noise makes it difficult to track the real interests of users in the session, while the inaccuracy of this low-level interest learning further reduces the learning effect of user interests at higher levels and affects the recommendation performance. Different from the above research, we divide the long sequence into subsequences in different time periods and use the improved graph attention network and LSTM to learn the complex structure within subsequences and the sequential dependencies among the subsequences, respectively. The graph attention network with node importance factor can fully extract the interests of users in different time periods, and LSTM can learn the dynamic changes of interests among different time periods. Unlike short-term interests, which are modeled based on sequential dependencies of recent interactions, long-term interests are modeled on long and complex sequences. Therefore, for long sequence, the strict order within subsequences not only has little effect on the entire sequence but also is prone to noise effects. And due to the complexity of long sequences, the use of graph attention network with node importance factor can effectively learn the importance of different items in the subsequence and the complex association between items, which reduces the noise effect and highlights the extraction of main interests.

For the fusion of long-term interests and short-term interests, Feng et al. [33] used the hyperparameter to control the addition of long-term and recent interests. However, the simple combination is difficult to capture the correlation of interests, and it is easy to make the model lose universality. Previous studies have shown that gating modules have more obvious advantages than simple concatenation or addition [15]. Tan et al. [2] proposed a dynamic memory-based attention network and used a gating module to adaptively adjust the importance of long-term and short-term interests. Tang et al. [16] proposed a mixture model M3, which fuses feature representations from different time scales based on the gating mechanism of Mixture-of-Experts (MOE). The above methods usually only consider the user's own information, ignoring the influence of neighbor users. Li et al. [18] proposed an FNUS model for finding similar neighbors from multiple perspectives, which divides the item set into three subspaces and searches for neighbor users from different subspaces, respectively. Banerjee et al. [19] used social network information to measure the correlation between users and combined it with scoring data and project characteristics. However, social network information is difficult to obtain. In order to further improve the recommendation performance, we integrate the neighbor features into the gating network and adaptively balance the influence of the users' short-term interests and long-term interests by aggregating neighbor information.

3. The Proposed Approach

Figure 1 shows the overall framework of the model proposed in this paper, which contains three main components, i.e., a

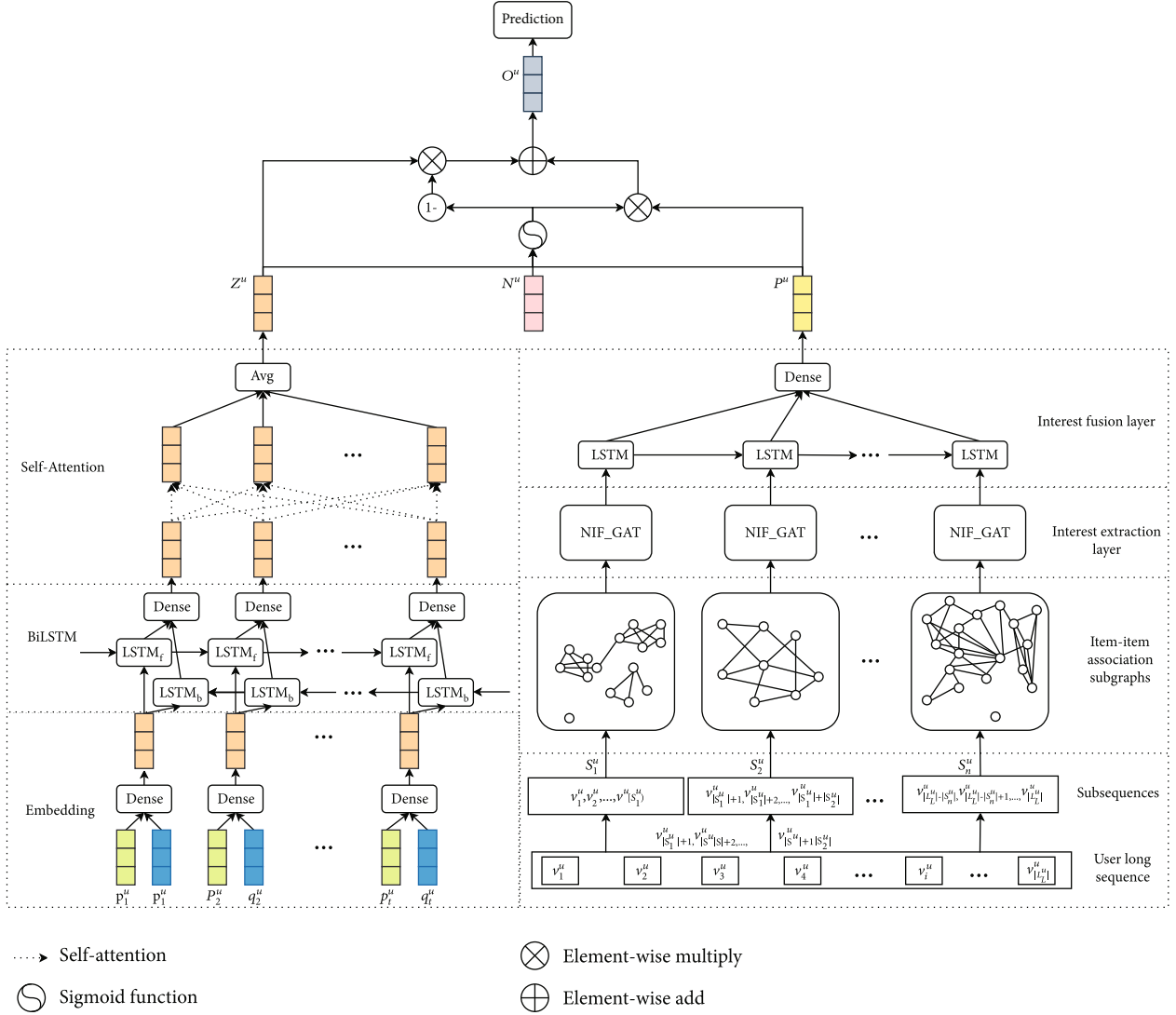


FIGURE 1: The overall framework of SRLIN.

short-term interest module based on BiLSTM and self-attention network, a long-term interest module based on the interest extraction layer and interest fusion layer, and a gated fusion module based on neighbor influence.

3.1. Notations and Problem Formulation. Let $U = \{u_1, u_2, \dots, u_m\}$ and $V = \{v_1, v_2, \dots, v_n\}$ denote the user set and the item set, respectively. For $\forall u \in U$, the behavior sequence of u refers to an ordered item set, which is sorted according to the interaction time of items selected by u in ascending order and denoted as $L^u = \{v_1^u, v_2^u, \dots, v_{|L^u|}^u\}$, where $v_i^u \in V$ represents the i th item on L^u . In order to facilitate interest extraction, the user behavior sequence is further divided into multiple subsequences if its sequence length exceeds the length threshold lenhrs or time span exceeds the time interval threshold Δt . Moreover, for two adjacent items on L^u , they are split into different subsequences if their time interval is greater than Δt or the length of the subsequence is greater than lenhrs . Let $S_k^u = \{v_{k,1}^u, v_{k,2}^u, \dots, v_{k,|S_k^u|}^u\}$ be the

k th subsequence on L^u , and S^u be the partition of L^u which is composed of all these subsequences. For $\forall u \in U$, let the latest subsequence before prediction time t be the short-term interest sequence L_S^u , and the set of subsequences on the behavior sequence be the long-term interest sequence L_L^u .

3.2. Item Embedding Based on Item Similarity and Dependency. Embedding is a common technique which transforms discrete values of data into numerical vectors that can be processed by the model. The neural network is usually used to convert sparse feature data into dense embeddings on the basis of one-hot embeddings. However, the method that only takes random initialization of item numbers tends to limit the focus of the model to historical records and ignores the potential relationship between items, which makes it difficult to capture the implicit interest features of users. In particular, on sparse datasets, it is difficult to achieve a good recommendation effect only by encoding the item numbers. Therefore, we learn item embeddings

from the perspectives of item similarity and dependency in the paper.

On the one hand, the item attribute information is the static features of the item itself, which can reflect the similarity between items. It shows that the corresponding feature vectors of items with the same attributes are also similar. Let \mathcal{F} be the item attribute set; we get one-hot embedding \mathbf{x}_i^f for $\forall f \in \mathcal{F}$ of item v_i and convert \mathbf{x}_i^f to dense embedding $\mathbf{p}_i^f \in \mathbb{R}^{d_f}$ according to the learnable embedding matrix \mathbf{W}_f . Next, these dense embeddings are concatenated to obtain the similarity embedding for item v_i , which is denoted as $\mathbf{p}_i \in \mathbb{R}^d$ and defined as follows:

$$\begin{aligned} \mathbf{p}_i^f &= \mathbf{W}_f \mathbf{x}_i^f, \\ \mathbf{p}_i &= \text{concat_func}\left(\left\{\mathbf{p}_i^f \mid f \in \mathcal{F}\right\}\right), \end{aligned} \quad (1)$$

where $\text{concat_func}(\cdot)$ means to connect the dense embedding \mathbf{p}_i^f of different attributes f .

On the other hand, inspired by association rules [34], we learn the item dependency embeddings from the global dependency of user history. Item dependencies not only reflect the similarity of users in their selection but also reflect the complementarity and cooccurrence relationship between items. For $\forall u \in U$, v_i and v_j on $L^u = \{v_1^u, v_2^u, \dots, v_{|L^u|}^u\}$ represent two dependent items. The item dependencies on all user history sequences are extracted, and a global dependency graph $\mathcal{G} = \langle V, \mathcal{E} \rangle$ is constructed accordingly. The global dependency graph is a directed graph. The nodes in the graph represent items, the directed edge $e_{ij} \in \mathcal{E}$ represents the successive clicks of items v_i to v_j , and the edge weight represents the cooccurrence degree between items, which is the number of times v_i to v_j appears repeatedly on different user sequences. Let $\mathbf{q}_i \in \mathbb{R}^d$ be the dependency embedding of item v_i on the global dependency graph \mathcal{G} , which is generated by using the Node2vec algorithm. By reflecting the characteristics of each node's neighbors through BFS, the probability of adjacent items appearing can be maximized.

Finally, the item embedding of $\forall v_i \in V$ is obtained by combining the two perspectives of the item similarity embedding \mathbf{p}_i based on static attributes and the dynamic item dependency embedding \mathbf{q}_i , which is defined as follows:

$$\mathbf{g}_i = \sigma(\mathbf{W}_g[\mathbf{p}_i, \mathbf{q}_i] + b_g), \quad (2)$$

where $\mathbf{W}_g \in \mathbb{R}^{d \times 2d}$ is the learnable weight matrix, σ is the *sigmoid* activation function, $\mathbf{g}_i \in \mathbb{R}^d$ is the item embedding of v_i , and d is the item embedding dimension.

3.3. Short-Term Interest Representation Based on BiLSTM and Self-Attention Network. Recurrent neural network [35] plays a prominent role in modeling sequential dependencies and is widely used in sequence recommendation, which can transmit and memorize the association among information to track the changing trend of user interests. Therefore, we make full use of the forgetting and remembering properties

of recurrent neural networks over time here. However, the simple RNN is difficult to deal with relatively long data due to its own structure, which makes it unable to meet the memory function of sequence data. In addition, considering that all user interactions in the recent period have an effect on predicting the choice of the next time. In order to take full advantage of the effect of different behaviors, we adopt BiLSTM to obtain the features of each time step in the users' recent sequences bidirectionally, which can model the dynamic changes of the users' short-term interests by mining the sequential dependencies of recent sequences.

The LSTM unit includes an input gate \mathbf{i}_t^u , a forgetting gate \mathbf{f}_t^u , an output gate \mathbf{o}_t^u , and a memory unit \mathbf{c}_t^u for state update. The calculation formulas are as follows:

$$\begin{aligned} \mathbf{i}_t^u &= \sigma(\mathbf{W}_{gi} \mathbf{g}_t^u + \mathbf{W}_{hi} \mathbf{h}_{t-1}^u + b_i), \\ \mathbf{f}_t^u &= \sigma(\mathbf{W}_{gf} \mathbf{g}_t^u + \mathbf{W}_{hf} \mathbf{h}_{t-1}^u + b_f), \\ \mathbf{o}_t^u &= \sigma(\mathbf{W}_{go} \mathbf{g}_t^u + \mathbf{W}_{ho} \mathbf{h}_{t-1}^u + b_o), \\ \tilde{\mathbf{c}}_t^u &= \tanh(\mathbf{W}_{gc} \mathbf{g}_t^u + \mathbf{W}_{hc} \mathbf{h}_{t-1}^u + b_c), \\ \mathbf{c}_t^u &= \mathbf{f}_t^u \mathbf{c}_{t-1}^u + \mathbf{i}_t^u \tilde{\mathbf{c}}_t^u, \\ \mathbf{h}_t^u &= \mathbf{o}_t^u \tanh(\mathbf{c}_t^u), \end{aligned} \quad (3)$$

where the input of LSTM is the item embeddings $\{\mathbf{g}_1^u, \mathbf{g}_2^u, \dots, \mathbf{g}_{|L_S^u|}^u\}$ in the recent sequence of user u and $|L_S^u|$ is the length of the recent sequence. The gate structure of LSTM is learned from the hidden state $\mathbf{h}_{t-1}^u \in \mathbb{R}^d$ of the previous output and the item embedding $\mathbf{g}_t^u \in \mathbb{R}^d$ of the current input, which is used to control the reception of current information, the memory of historical information, and selective output features. σ is the *sigmoid* activation function, \mathbf{W}_{gi} , \mathbf{W}_{hi} , \mathbf{W}_{gf} , \mathbf{W}_{hf} , \mathbf{W}_{go} , \mathbf{W}_{ho} , \mathbf{W}_{gc} , \mathbf{W}_{hc} $\in \mathbb{R}^{d \times d}$ are learnable weight matrices, and b_i , b_f , b_o , b_c are biases.

BiLSTM includes forward and backward LSTM. They have the same structure and the same input data, but the direction of the sequence input is different. At time t , BiLSTM can be expressed as

$$\mathbf{h}_{ct}^u = [\mathbf{h}_{ft}^u, \mathbf{h}_{bt}^u], \quad (4)$$

where \mathbf{h}_{ft}^u is the output of the forward LSTM, \mathbf{h}_{bt}^u is the output of the backward LSTM, and $\mathbf{h}_{ct}^u \in \mathbb{R}^{2d}$ is obtained by connecting \mathbf{h}_{ft}^u and \mathbf{h}_{bt}^u .

We input \mathbf{h}_{ct}^u into a fully connected layer, and the resulting $\mathbf{H}_t^u \in \mathbb{R}^d$ is regarded as the output of the BiLSTM layer at time t :

$$\mathbf{H}_t^u = \sigma(\mathbf{W}_H \mathbf{h}_{ct}^u + b_H), \quad (5)$$

where $\mathbf{W}_H \in \mathbb{R}^{d \times 2d}$ is the learnable weight matrix and σ is the *sigmoid* activation function.

However, there may be random or accidental behaviors in the recent sequence of users, which affect the learning effect of the recurrent neural network on the users' interests

and deviate from the users' true intention. Different from the recurrent neural network, the attention network regards the input content as a whole, which alleviates the influence of noise by assigning higher weights to the important interests of users. Therefore, on the basis of using BiLSTM to model the sequential dependencies of users' short-term interests, we use the self-attention network to amplify the key parts of users' short-term interests that are conducive to prediction.

To further extract the important information of users' short-term interest representations, we input $\mathbf{H}^u = \{\mathbf{H}_1^u, \mathbf{H}_2^u, \dots, \mathbf{H}_{|L_S^u|}^u\}$ into the self-attention network.

The self-attention network can be described as

$$\mathbf{Z}_{L_S^u}^u = \text{attention}(\mathbf{Q}^u, \mathbf{K}^u, \mathbf{V}^u) = \text{soft max} \left(\frac{\mathbf{Q}^u (\mathbf{K}^u)^T}{\sqrt{d}} \right) \mathbf{V}^u, \quad (6)$$

where $\mathbf{Z}_{L_S^u}^u \in \mathbb{R}^{|L_S^u| \times d}$ is the output of the self-attention network, $|L_S^u|$ is the recent sequence length, $\mathbf{Q}^u, \mathbf{K}^u, \mathbf{V}^u \in \mathbb{R}^{|L_S^u| \times d}$ are obtained from the input \mathbf{H}^u through the linear transformation of the weight matrices $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{d \times d}$, respectively, that is, $\mathbf{Q}^u = \mathbf{H}^u \mathbf{W}_Q$, $\mathbf{K}^u = \mathbf{H}^u \mathbf{W}_K$, and $\mathbf{V}^u = \mathbf{H}^u \mathbf{W}_V$.

Finally, short-term interest representation $\mathbf{Z}^u \in \mathbb{R}^d$ of user u is calculated by average pooling on $\mathbf{Z}_{L_S^u}^u$:

$$\mathbf{Z}^u = \text{Avg}(\mathbf{Z}_{L_S^u}^u). \quad (7)$$

3.4. Long-Term Interest Representation Based on Interest Extraction Layer and Interest Fusion Layer. As the user's long-term behavior sequence often contains noise and has a relatively large time span, it is difficult to directly model the overall long-term sequence, resulting in unsatisfactory recommendation results. Therefore, we divide the long sequence into multiple subsequences. Each subsequence reflects the user's interests over a period of time. By using the hierarchical mechanism of interest extraction layer and interest fusion layer, the different associations between items in subsequences and the sequential dependencies among subsequences are modeled to jointly generate long-term interest representations of user. The interest extraction layer can effectively extract the main interests in the subsequences, and the interest fusion layer can dynamically learn the order changes of user interests among different subsequences.

3.4.1. Interest Extraction Layer. Each subsequence corresponds to a time period. Users have different interests in different periods and may also have multiple interests in the same period. In order to highlight the important parts that affect prediction in different time periods, we use the graph attention network with node importance to extract the main interests in different subsequences, respectively.

The graph attention network is a graph neural network combined with attention mechanism, which uses self-

attention to learn the graph structure and has efficient parallel computing capabilities. The update of the feature of each node in the graph relies on the attention calculation of its neighbor nodes, which is realized by assigning different weights to the neighbor nodes.

Different from the simple sequential structure, the graph attention network can more clearly model the complex correlation between items. By analyzing the internal structure of the item graph, the more complex and implicit connections between user clicks can be captured. Unlike the features of each item in the recent sequence, which have an impact on the learning of short-term interests, the graph attention network does not consider the sequential association between items. The reason is that the main interests of the subsequences are more emphasized in the subsequence interest extraction stage. For example, in the online shopping system, users may have multiple needs at the same time, and the purchase order may be "a shirt, a bunch of flowers, a basket of apples, a bunch of bananas, a vase." The relationship between "flower" and "vase" should be closer, but it is interrupted by "apple" and "banana" in the actual purchase. If we follow the strict order of the subsequences, it is difficult to extract the main interest of the subsequence. Therefore, the sequential relationship within the subsequence is a negative effect on the long sequence composed of multiple subsequences. It not only reduces the model efficiency but also easily affects the recommendation effect.

In addition, since there may be behaviors deviating from the user's interests in the subsequences, adopting a graph attention network can further reduce the influence of noise while learning the relationship between items. The reason is that the attention mechanism can amplify the features that are helpful for decision-making and ignore unimportant or irrelevant information.

It can be said that the graph attention network can model the complex relationship between different items, automatically learn the important features in the graph, and suppress the influence of noise. The attention mechanism enables the graph structure to better achieve neighbor aggregation, and the graph structure also provides a degree of interpretability for the attention mechanism.

For any subsequence S_k^u on the long sequence L_L^u , the items of the subsequence S_k^u are regarded as nodes. Considering that almost no items in the short sequence are clicked repeatedly, we use the similarity between items to describe the connection relationship between nodes. In this way, the subsequence data $S_k^u = \{v_{k,1}^u, v_{k,2}^u, \dots, v_{k,|S_k^u|}^u\}$ is transformed into a subgraph structure $G_k^u = \langle V_k^u, E_k^u \rangle$, where item $v_{k,i}^u$ is represented as node $i \in V_k^u$. For the sake of simplicity, there exists an edge between two item nodes only if their similarity is more than 0. The input data of the graph attention network is the node features $\{\mathbf{g}_{k,1}^u, \mathbf{g}_{k,2}^u, \dots, \mathbf{g}_{k,|S_k^u|}^u\}$, where $\mathbf{g}_{k,i}^u \in \mathbb{R}^d$ represents the feature of the i th node in the subsequence and $|S_k^u|$ is the length of the subsequence.

In the graph attention network, the importance of the neighbor node j to the current node i is defined as

$$e_{ij}^u = \text{LeakyReLU} \left(\mathbf{W}_a^T \left[\mathbf{W}_g \mathbf{g}_{k,i}^u \parallel \mathbf{W}_g \mathbf{g}_{k,j}^u \right] \right), \quad (8)$$

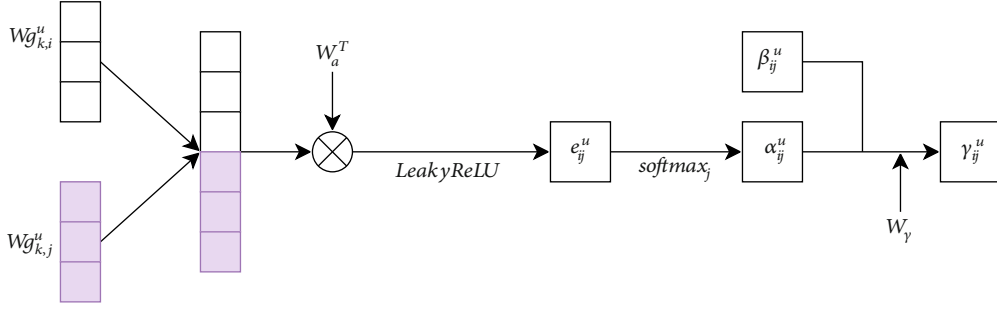


FIGURE 2: Attention coefficients.

where $\mathbf{W} \in \mathbb{R}^{d \times d}$ represents a learnable shared weight matrix which is used to improve the expressive ability of item feature $\mathbf{g}_{k,i}^u$ and \parallel represents connecting the features of node i and node j . The attention function is a single-layer feedforward neural network. $\mathbf{W}_a \in \mathbb{R}^{2d}$ is used to learn the influence of node j on node i , and LeakyReLU denotes nonlinear activation.

The attention coefficient is calculated by considering the first-order aggregation of all neighbor nodes on node i , which is defined as follows:

$$\alpha_{ij}^u = \text{softmax}_j(e_{ij}^u) = \frac{\exp(e_{ij}^u)}{\sum_{x \in N_i} \exp(e_{ix}^u)}, \quad (9)$$

where N_i is the neighbor set of node i (including i).

The feature vector of node i can be obtained by applying the attention coefficients to the corresponding neighbors of node i and combining the features, which is denoted as

$$\tilde{\mathbf{g}}_{k,i}^u = \sigma \left(\sum_{j \in N_i} \alpha_{ij}^u \mathbf{W} \mathbf{g}_{k,j}^u \right). \quad (10)$$

The weights of nodes in the graph attention network are learned through weighted aggregation of their neighboring nodes. However, due to the normalization of *softmax*, the importance attribute of nodes in the whole graph is not better highlighted. For subsequences, the importance of different item nodes has a great influence on the extraction of main interests. Generally speaking, the more adjacent nodes a node is associated with, the higher the importance of the node. Therefore, we improve the attention coefficient of the graph attention network to more clearly distinguish the importance of different nodes.

The calculation process of the new attention coefficient is shown in Figure 2.

In Figure 2, the importance of nodes is reflected by the degree of nodes. By normalizing the degree values of all nodes in the subsequence, the importance β_i^u of any node i can be obtained. $\mathbf{W}_\gamma \in \mathbb{R}^{1 \times 2}$ means that the information of node importance β_i^u and attention coefficient α_{ij}^u is fused to obtain a new attention coefficient γ_{ij} .

$$\gamma_{ij}^u = \mathbf{W}_\gamma \left[\alpha_{ij}^u, \beta_i^u \right]. \quad (11)$$

After applying the new attention coefficient γ_{ij}^u to the feature combination, the final output features $\hat{\mathbf{g}}_{k,i}^u \in \mathbb{R}^d$ of node i is obtained.

$$\hat{\mathbf{g}}_{k,i}^u = \sigma \left(\sum_{j \in N_i} \gamma_{ij}^u \mathbf{W} \mathbf{g}_{k,j}^u \right). \quad (12)$$

$\{\hat{\mathbf{g}}_{k,1}^u, \hat{\mathbf{g}}_{k,2}^u, \dots, \hat{\mathbf{g}}_{k,|S_k^u|}^u\}$ denotes the output of the subsequence data through the graph attention network.

Finally, the output is aggregated into a vector $\mathbf{r}_k^u \in \mathbb{R}^d$ by average pooling, which represents the main interests of the k th subsequence.

3.4.2. Interest Fusion Layer. In the interest fusion layer, the subsequence feature $\mathbf{r}_k^u \in \mathbb{R}^d$ of each time period is regarded as the basic unit, and we use LSTM to learn the sequential dynamic changes of user interests among different time periods because there is a relative order among different periods. In period k , the LSTM unit can be abbreviated as

$$\hat{\mathbf{h}}_k^u = \text{LSTM}(\mathbf{r}_k^u, \hat{\mathbf{h}}_{k-1}^u), \quad (13)$$

where $\{\mathbf{r}_1^u, \mathbf{r}_2^u, \dots, \mathbf{r}_{|S^u|}^u\}$ is the feature set of subsequences obtained through the graph attention network, $|S^u|$ is the number of subsequences, and $\hat{\mathbf{h}}_{k-1}^u \in \mathbb{R}^d$ is the output of the previous period.

The outputs of all time steps of LSTM are fused to obtain the long-term interest representation $\mathbf{P}^u \in \mathbb{R}^d$ of user u :

$$\hat{\mathbf{H}}^u = \left\| \left\|_{k=1}^{|S^u|} \hat{\mathbf{h}}_k^u \right\| \right\|, \quad (14)$$

$$\mathbf{P}^u = \sigma(\mathbf{W}_p \hat{\mathbf{H}}^u + b_p),$$

where $\left\| \left\| \right\|$ means to connect the outputs of different subsequences of LSTM and σ is the *sigmoid* activation function.

3.5. Gating Fusion Mechanism Based on Neighbor Influence. The interests of users change dynamically over time and the degree of change varies for different users, which indicates that the long-term and short-term interests of different users

have different degrees of influence on their interest predictions [36]. However, in addition to relying on their own interests, user intentions may also be affected by their neighbors. Thus, we introduce the gating network to fuse the user's own interests and neighbor features, which adaptively adjust the weights of long-term and short-term interest features by considering the influence of neighbors.

In a real system, users may keep some of their attributes secret out of privacy; that is, users may obscure information such as gender and age. Considering the defect of incomplete user attribute information, we learn the nearest neighbor representations of users from the perspective of historical behavior data. First, for each item, all users who interact with the item is regarded as a text, and each user is regarded as a word in the text. The user's word embedding is obtained by using the word2vec algorithm. Then, these user embeddings are clustered by using the K -means [37] algorithm, where the number of clusters K is determined according to the elbow method. Finally, the average of user embedding in each class is calculated and denoted as the nearest neighbor embedding $\mathbf{N}^u \in \mathbb{R}^d$.

The user's short-term interest representation $\mathbf{Z}^u \in \mathbb{R}^d$, the long-term interest representation $\mathbf{P}^u \in \mathbb{R}^d$, and the nearest neighbor representation \mathbf{N}^u are used as the input, and the gating network is designed as follows:

$$\mathbf{G}^u = \sigma(\mathbf{W}_G[\mathbf{Z}^u, \mathbf{P}^u, \mathbf{N}^u] + b_G), \quad (15)$$

where $\mathbf{W}_G \in \mathbb{R}^{d \times 3d}$ is the weight matrix of linear transformation, σ is the *sigmoid* activation function, and the gate vector is $\mathbf{G}^u \in \mathbb{R}^d$.

Finally, by adaptively allocating the proportion of long-term and short-term interest through the gate vector, the user interest representation vector is obtained, which is denoted as \mathbf{O}^u and defined as follows:

$$\mathbf{O}^u = \mathbf{G}^u \odot \mathbf{Z}^u + (1 - \mathbf{G}^u) \odot \mathbf{P}^u, \quad (16)$$

where \odot denotes element-wise multiplication.

3.6. Model Optimization. To obtain the user's recommendation list, the predicted probability distribution is generated through the softmax layer, and the cross-entropy is used as the loss function to train the predicted probability of the target item v_{tar}^u . Considering the large number of items in the real system and the high computational cost, we use sampled-softmax [2] to speed up training, which is defined as follows:

$$\mathcal{L} = - \sum_{\mathcal{N}} \log \frac{\exp\left(\left(\mathbf{g}_{\text{tar}}^u\right)^T \mathbf{O}^u\right)}{\sum_{j \in \mathcal{N}} \exp\left(\mathbf{g}_j^T \mathbf{O}^u\right)}, \quad (17)$$

where \mathcal{N} is the sampling subset selected from the item set V according to the sampling function, $\mathbf{g}_{\text{tar}}^u$ is the embedding vector of the target item v_{tar}^u , and \mathbf{O}^u is the user interest vector.

4. Experimental Results and Analysis

4.1. Datasets and Parameter Settings. We conduct experiments on two public datasets, i.e., MovieLens 1M dataset and JD dataset.

- (1) MovieLens 1M: the MovieLens dataset is a rating dataset provided by the GroupLens group of the Minnesota Computer Institute, which includes user statistics, movie information, rating time, and rating values. The MovieLens 1M dataset contains 1,000,209 ratings of 3,952 movie items by 6,040 users, and each user has rated at least 20 items. The higher the user's rating on an item, the more the user likes it
- (2) JD: the JD dataset records the user shopping behavior data of the JD e-commerce operation platform from February 1, 2018, to April 15, 2018. It is a relatively sparse dataset with a relatively short time span, which contains 37,214,269 records of 378,457 commodity items by 1,608,707 users

On the datasets, the user history records are sorted by time, and each sequence is divided into multiple subsequences according to the division rules. The data is preprocessed with reference to the experimental setting of the model DMAN [2], the last and penultimate interactions of each user are used as testing and validation, respectively, and the rest is used for training. We run five experiments repeatedly and take the average of the five results as the experimental results.

The model is optimized using Adam with a learning rate of 0.001, and the batch size is 512. In order to ensure the consistency of the experiments, the item embedding dimensions are set to 128. In the gating network based on the influence of neighbors, the number of user clusters k is determined according to the elbow method. To quickly and effectively determine the range of neighbors, the elbow method is used to calculate the degree of distortion. The degree of distortion is the sum of the squared errors (SSE) of the distance between the particle in each class and the sample points in the class. By constructing the distortion degree image of the number of clusters, the elbow position with the most obvious distortion degree change is taken as the best cluster k value. Figures 3 and 4 show the variation of the distortion degree SSE with the cluster k value on the MovieLens 1M and JD datasets, respectively. It can be seen from Figure 3 that the change of SSE is most obvious when k value is between 25 and 50 on MovieLens 1M dataset; we set the number of user clusters k to 40. In Figure 4, when the cluster k value on the JD dataset is around 25, the change of SSE is the most obvious. Therefore, the number of user clusters k is set to 25 on the JD dataset.

4.2. Comparison Methods. In order to verify the effectiveness of the proposed model, the following methods are selected for experimental comparison:

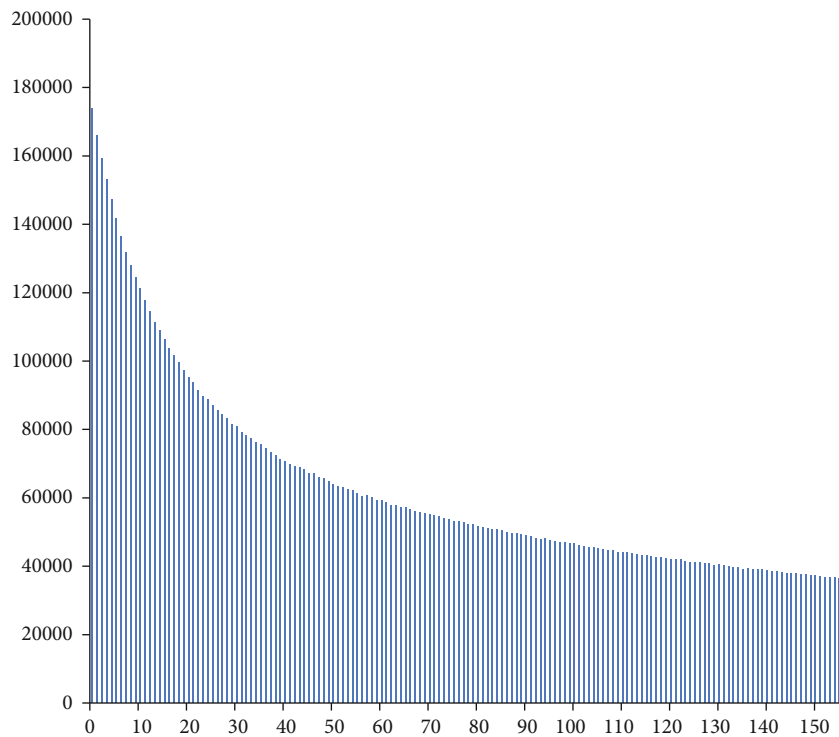


FIGURE 3: SSE of different number of k on MovieLens 1M.

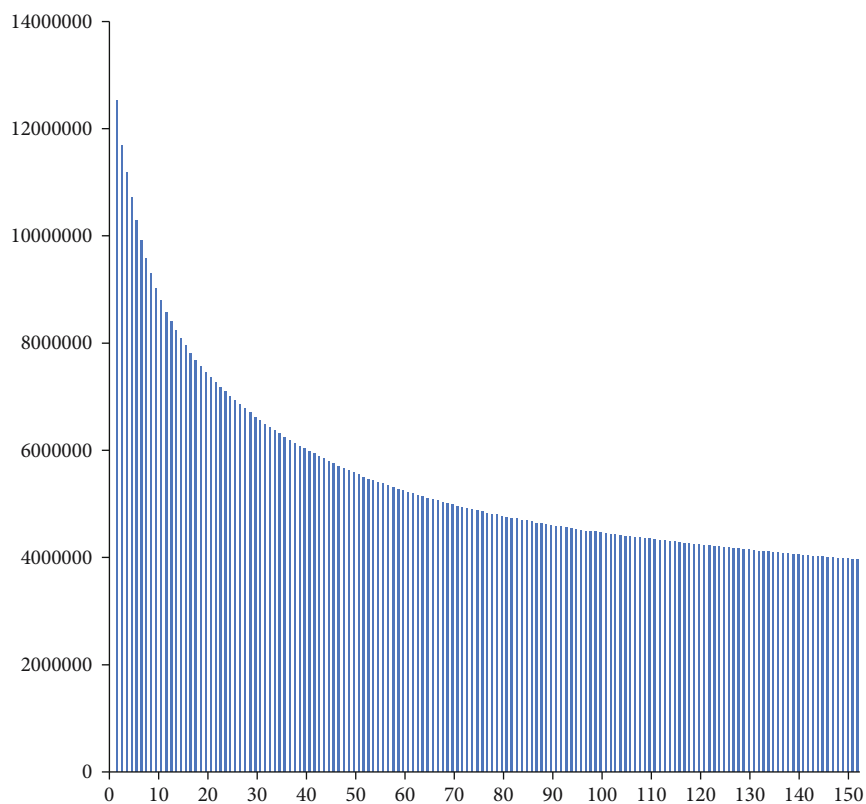


FIGURE 4: SSE of different number of k on JD.

- (1) GRU4Rec+ [8]: GRU4Rec [38] is a classic RNN-based model for session recommendation. Based on GRU4Rec, GRU4Rec+ proposes a new ranking loss function and improves the sampling strategy
- (2) Caser [7]: it models the user’s recent behaviors as an “image” based on time and latent features and learns the image through CNN. Applying both horizontal and vertical convolutional filters to image learning can capture complex features such as point-level, union-level sequence patterns, and skipping behaviors in sequences
- (3) SASRec [9]: SASRec is a neural network model composed of stacked self-attention which uses the self-attention mechanism to assign different weights to sequence data and learn more complex feature transformations through the hierarchical network
- (4) SHAN [30]: SHAN is a sequential recommendation method based on hierarchical attention network. The first layer learns the user’s long-term interests, and the second layer comprehensively considers the user’s long-term interests and short-term interests. Both layers of attention network use user embedding vector as attention query for interest learning, which realizes personalized recommendation
- (5) SDM [15]: SDM is a novel sequence deep matching model which used the multihead self-attention mechanism to obtain the recent diverse interests of users and learned the long-term interests of users by modeling long-term features of different categories. In this model, according to the obtained user’s personalized interests, a gating module is used to fuse the short-term interest related parts of the complex and diverse long-term interests
- (6) DMAN [2]: the DMAN model designs the recursive self-attention network to model users’ short-term interests and preserves the important content of long-term interests as much as possible by maintaining a set of dynamically updated memory blocks. This model also used a gating network to combine long-term and short-term interests for recommendation

4.3. Comparison Methods. In order to evaluate the recommendation performance of different methods, we use hit rate (HR@K) and normalized discounted cumulative gain (NDCG@K) as evaluation metrics.

HR@K (Hit Rate@K) represents the percentage of items in the top-K of the ranking list in the test case, which is used to measure the accuracy of recommendation and defined as

$$\text{HR@K} = \frac{1}{N} \sum_{i=1}^K \text{hit}(i), \quad (18)$$

where N is the number of test cases and $\text{hit}(\cdot)$ is the indicator function. $\text{hit}(\cdot) = 1$ means that the item selected by the

user appears in the top-K recommendation list; otherwise, $\text{hit}(\cdot) = 0$.

NDCG@K (normalized discounted cumulative gain (NDCG)) is an evaluation metric about ranking. The higher the ranking of the correctly recommended item, the better the recommendation effect and the higher the NDCG value. This metric considers the order of recommendation results and denoted as

$$\text{NDCG} = \frac{1}{N} \sum_{n \in N} \frac{\text{DCG}}{\text{IDCG}}. \quad (19)$$

DCG is the cumulative gain of loss and defined as

$$\text{DCG} = \sum_{i=1}^K \frac{2^{r_i} - 1}{\log_2(i + 1)}, \quad (20)$$

where r_i is the correlation of the item at position i . If the recommended item is in the test case, $r_i = 1$; otherwise, $r_i = 0$. The higher the ranking of related items, the higher the value of DCG. IDCG is the DCG that rearranges the items in the recommendation list according to their relevance. Considering that the DCG values of different users may vary greatly, IDCG is used to normalize the DCG of different users to obtain the evaluation metric NDCG.

4.4. Comparison with Baseline Methods. Table 1 lists the experimental results of our model and six baselines on MovieLens 1M and JD datasets, where the bold ones represent the best results and the underlined ones are the second best results.

As listed in Table 1, we can make the following observations.

- (1) For GRU4Rec+, Caser, and SASRec, which focus on short-term interest modeling, they do not perform well on two experimental datasets. GRU4Rec+ has the worst recommendation, which may be because the recurrent neural network of sequential modeling cannot effectively deal with the interest offset behaviors in the sequences and is easily affected by noise. Caser has better results due to considering more user personalization information. The performance of SASRec is significantly better than that of GRU4Rec+ and Caser. On the one hand, it shows that the attention network with position bias is beneficial to extract users’ dynamic interests and alleviate the influence of noise. On the other hand, it shows that the stacked hierarchical attention network has significant advantages in dynamic modeling, which also explains the effectiveness of our model using a hierarchical structure
- (2) For SHAN, SDM, and DMAN, which consider longer interaction sequences, the recommendation performance of these models is generally higher than those of the short-term interest models, i.e., GRU4Rec+, Caser, and SASRec. These results show that the long-term interests of users are also important for predicting users’ choice. Therefore,

TABLE 1: Performance evaluation of different recommendation models (%).

Models	MovieLens 1M			JD		
	HR@10	HR@50	NDCG@100	HR@10	HR@50	NDCG@100
GRU4Rec+	17.69	43.13	16.90	27.65	38.73	23.40
Caser	18.98	45.64	17.62	29.27	40.16	24.25
SASRec	21.02	47.28	19.05	33.98	44.89	27.41
SHAN	21.34	49.52	19.55	37.72	50.55	29.80
SDM	23.42	51.26	20.44	40.68	55.30	34.82
DMAN	<u>25.18</u>	<u>53.24</u>	<u>22.03</u>	<u>44.58</u>	<u>58.82</u>	36.93
SRLIN	26.97	55.84	22.66	44.97	59.44	<u>34.96</u>

considering long-term interests based on short-term interests modeling can further improve the performance of recommendation

- (3) For SHAN, SDM, and DMAN, which consider long-term interests and short-term interests, the recommendation results of SDM consistently outperform those of SHAN in evaluation metrics HR@10, HR@50, and NDCG@100. This is mainly due to the difference between the two models in the fusion of long-term and short-term interests. SHAN adopts a hierarchical attention network to integrate long-term and short-term interests, while SDM adopts a gating network. The gating network is more effective than the hierarchical attention network for learning the interest expression. In addition, the attribute feature extraction of the input data by SDM further improves the expressive ability of the model. DMAN can achieve better recommendation results than SDM because DMAN employs a dynamic memory-based attention network to continuously aggregate long-term representations into a set of memory blocks. By dividing subsequences, complex problems can be simplified. It is easier and more effective than SDM that directly extracts interests from the whole long-term sequence and can better express the users' long-term interest features

- (4) For our proposed model SRLIN, it shows excellent recommendation results on both datasets. Compared with SDM, SRLIN has an average improvement of 3.92% in HR@10, 4.36% in HR@50, and 1.18% in NDCG@100. Compared with DMAN, SRLIN improves by 1.79% and 0.39% in HR@10, and 2.6% and 0.62% in HR@50, respectively. The overall effectiveness of our model can be attributed to several aspects. First, embedding representations of items are learned from multiple perspectives, which helps alleviate data sparsity issues. Second, in the long-term interest modeling, the graph attention network with node importance is used to learn the main features of the subsequences, which can not only accurately and fully extract stable changing long-term interests but also effectively eliminate the noise influence in the subsequences. Third, the long-term and short-term interests of users are comprehensively considered, and the interests are fused through the gating network

together with the neighbor user information. The application of neighbor user features makes the model consider the influence of neighbor information while focusing on the user's own personalized data, which can enrich the prediction of user intention and improve the recommendation performance

- (5) It is noted that the SRLIN model can achieve the best recommendation effect in the metric of NDCG@100 on the MovieLens 1M dataset, while the experimental result on the JD dataset is suboptimal. This is because the time span of the JD dataset is relatively short and the average sequence length of users is not long, which makes it difficult to fully learn stable changing long-term interests when modeling long-term representations. Comparing the HR@K metrics on the two datasets, we find that SRLIN achieves the average improvement of 1.09% on HR@10 and 1.61% on HR@50. It shows that the recommendation effect of the SRLIN model can be improved compared with the baselines as the length of the recommendation list increases, which further explains the reason why the ranking metric NDCG of SRLIN on the JD dataset is not the best

4.5. Effect of Graph Attention Network with Node Importance Factor. To explore the advantages of SRLIN using graph attention network with node importance factor in the interest extraction layer, we design three additional variants, i.e., SRLIN-RNN, SRLIN-AT, and SRLIN-GAT.

- (1) SRLIN-RNN: LSTM is used to learn the interests of subsequences. Because of the order-dependent property of LSTM itself, the order relationship within subsequences is considered when extracting interests
- (2) SRLIN-AT: the attention network is used to learn the interests of subsequences, and the attention mechanism can capture the main features of subsequences
- (3) SRLIN-GAT: the main interests of subsequences are learned using graph attention network without considering the importance of nodes

Table 2 lists the experimental results in the evaluation metric of HR@50 for different subsequence interest extraction methods on MovieLens 1M and JD datasets.

By observing the results in Table 2, the following can be found:

- (1) The SRLIN-RNN method, which uses LSTM to learn subsequence interests, performs the worst among the four models. This is because the interest offset caused by random, combined, jumping, and other behaviors in the historical sequences, which makes the recurrent neural network susceptible to noise when modeling subsequence sequential dependencies. The information loss of the bottom layer LSTM can further affect the learning of the upper layer interest changes, resulting in poor recommendation effect
- (2) The performance of SRLIN-AT with attention network is better than that of SRLIN-RNN. The reason is that the attention mechanism pays more attention to important interest features, which alleviates the noise effect caused by interest offset in subsequences to a certain extent
- (3) SRLIN-GAT uses graph attention network to extract the main interests of subsequences and can obtain better results than SRLIN-AT, which shows the effectiveness of modeling complex associations of items. The graph structure of the graph attention network visually draws the neighbor aggregation of items, which can capture more implicit connection relationship between items and help the attention mechanism to extract the main interests
- (4) These results show that our SRLIN model outperforms the three variants. The reason why SRLIN is better than the best variant SRLIN-GAT is the introduction of the importance of item nodes. It shows that in a period of time, the importance of different items has a strong impact on user interests, reflecting users' different degrees of preferences. The more the number of nodes related to a node, the higher the importance of the node, and the contribution of the node to the subsequence is also greater. Therefore, considering the importance of different items in the subsequence is beneficial to the extraction of the main interests

4.6. Effects of Individual Components. To verify the effectiveness of each part of the model, we design two additional variants, i.e., SRLIN-S and SRLIN-G. SRLIN-S removes the long-term interest modeling module of SRLIN, while the gating module of SRLIN-G only considers the user's long-term and short-term interests.

Table 3 lists the experimental results in the evaluation metric HR@50 for the three methods on MovieLens 1M and JD datasets.

By analyzing the experimental results in Table 3, we find that the experimental results of interest fusion models SRLIN-G and SRLIN are always significantly better than those of SRLIN-S, which indicates the effectiveness of modeling long-term interest representations for recommen-

TABLE 2: Comparison of hit rate for different interest extraction methods (%).

Models	MovieLens 1M	JD
SRLIN-RNN	49.15	51.16
SRLIN-AT	53.02	54.64
SRLIN-GAT	55.66	56.18
SRLIN	55.84	59.44

TABLE 3: Comparison of hit rate for three methods with different components (%).

Models	MovieLens 1M	JD
SRLIN-S	46.56	42.30
SRLIN-G	54.83	57.41
SRLIN	55.84	59.44

TABLE 4: Comparison of hit rate for two methods with different embedding methods (%).

Models	MovieLens 1M	JD
SRLIN-RD	50.15	51.42
SRLIN	55.84	59.44

dation results. The user interest information carried by long-term interest representation and short-term interest representation plays an important role in the recommendation. They complement and correlate with each other, which can further improve recommendation performance. In addition, compared with SRLIN-G, our SRLIN model can capture the influence of neighbor user feature, which makes it achieve better recommendation effect. These results show that the gating network considering the neighbor features can better balance the users' long-term interests and short-term interests so that it can obtain more accurate user interest representations.

4.7. Effect of Item Embeddings from Multiple Perspectives. To show the recommendation effect of different item embedding methods, we design an additional variant SRLIN-RD, which randomly encodes item embeddings based on item numbers. We compare the variant with our SRLIN that fuses item embeddings from multiple perspectives and validate them using the evaluation metric HR@50. The experimental results are shown in Table 4.

It can be seen from Table 4 that SRLIN has significant advantages. In contrast, the performance of SRLIN-RD is significantly reduced. In particular, on the JD dataset, the effect of SRLIN-RD is much lower than that of SRLIN, because the JD dataset has higher sparsity than the MovieLens 1M dataset. These experimental results show that learning item embeddings from multiple perspectives can effectively alleviate the problem of data sparsity, thereby improving recommendation performance.

4.8. Effect of Time and Length Thresholds. The time interval threshold Δt and the subsequence length threshold l_{thrs}

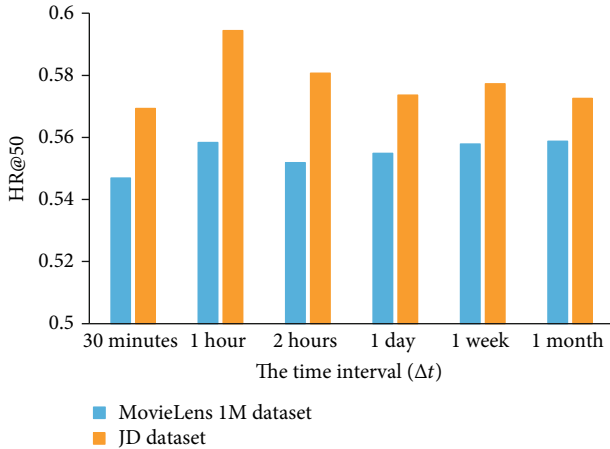


FIGURE 5: Effect of different time interval threshold Δt on hit rate.

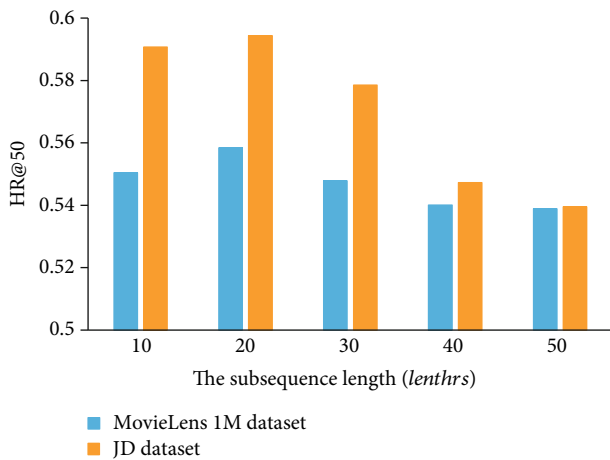


FIGURE 6: Effect of different length threshold len_{thrs} on hit rate.

are used in the sequence division. We verify the influence of different time interval sizes and sequence lengths on the recommendation performance through experiments. In order to intuitively show the effect of different thresholds, one parameter is fixed to verify the effect of another parameter, and the following experiments are designed for comparative analysis.

For a user history sequence, the sequence is divided if the time interval between adjacent items is more than the threshold Δt . The time interval Δt is used to distinguish the interests of users in different time periods. First, we fix the subsequence length threshold $len_{thrs} = 20$ and set the time interval threshold Δt to different values. The experimental results are shown in Figure 5.

By analyzing the results in Figure 5, it can be seen that the experimental result is optimal when the time interval threshold is set to 1 hour on the JD dataset, which indicates that most users choose products compactly within a period of time. When the time interval exceeds 1 hour, the occurrence of the next behavior is highly likely to indicate that the user reopens JD page, and the user may have new needs at this time. On the MovieLens 1M dataset, we find that the time interval has little effect on the recommendation results.

This is because user preferences tend to be stable in movie selection, and the time interval does not clearly distinguish user interest changes. For uniformity, we set the time interval threshold Δt to 1 hour on both datasets.

Then, in order to more accurately reflect the main interests of users in a period of time, we further divide the subsequences that meet the time interval requirements. When the subsequence length exceeds the threshold len_{thrs} , it is considered that the user has started the next selection. Figure 6 shows the effect of the length threshold len_{thrs} on the recommendation effect when the time interval is set to 1 hour. As shown in Figure 6, the hit rate decreases as the subsequence length increases when the subsequence length exceeds 20. These results show that the user's demand is usually determined within 20 selection items. The observation also illustrates that the user's interests change dynamically over time. Therefore, we set the subsequence length threshold to 20.

5. Conclusions

In this paper, we propose a sequential recommendation model for long-term interest memory and nearest neighbor influence. The model learns item embeddings from multiple perspectives, which alleviates the problem of data sparsity by capturing the implicit relationship between items. For the case of long and complex behavior sequences of users, a hierarchical processing method is introduced to capture users' long-term interests by modeling complex structure within subsequences and sequential dependencies among subsequences, which deals with the problem of imperfect long-term interests modeling. In the interest extraction layer, we design the graph attention network with node importance factors which can fully learn the importance of different items in the subsequence and the complex relationship between the items and can focus on the important interests of each subsequence. In addition, we also design a gating network that considers the features of user neighbors. It comprehensively learns the relationship among each user's neighbor representation, long-term interest representation, and short-term interest representation, so as to solve the inadequacy of user interest prediction only relying on its historical behaviors. Extensive experiments on the MovieLens 1M and JD datasets show that our model outperforms baselines in prediction performance.

On the JD dataset, many user sequences are short or have a short time span, in which it is not suit for learning the long-term stable interests. Therefore, in the future, we will further explore the latent features of long-term interests and strive to reduce the time cost of the model. In addition, knowledge graph can provide more relevant information, which is also worthy of further consideration.

Data Availability

The data used to support the findings of this study can be downloaded from <https://grouplens.org/datasets/movielens/> and <https://jddata.jd.com/html/detail.html?id=8>.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work is funded by the Science and Technology Project of Hebei Education Department (ZD2022105), the Natural Science Foundation of Hebei Province, China (F2020201023), and the high-level personnel starting project of Hebei University (521100221089).

References

- [1] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "Lightgcn: simplifying and powering graph convolution network for recommendation," in *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pp. 639–648, Virtual Event China, 2020.
- [2] Q. Tan, J. Zhang, N. Liu et al., "Dynamic memory based attention network for sequential recommendation," 2021, <http://arxiv.org/abs/2102.09269>.
- [3] B. Paudel and A. Bernstein, "Random walks with erasure: diversifying personalized recommendations on social and information networks," in *Proceedings of the Web Conference 2021*, pp. 2046–2057, Ljubljana Slovenia, 2021.
- [4] Z. He, C. Y. Chow, and J. D. Zhang, "GAME: learning graphical and attentive multi-view embeddings for occasional group recommendation," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 649–658, Virtual Event China, 2020.
- [5] S. Wang, L. Hu, Y. Wang, L. Cao, Q. Z. Sheng, and M. Orgun, "Sequential recommender systems: challenges, progress and prospects," 2019, <http://arxiv.org/abs/2001.04830>.
- [6] S. Wang, L. Cao, Y. Wang, Q. Z. Sheng, M. A. Orgun, and D. Lian, "A survey on session-based recommender systems," *ACM Computing Surveys (CSUR)*, vol. 54, no. 7, pp. 1–38, 2022.
- [7] J. Tang and K. Wang, "Personalized top-n sequential recommendation via convolutional sequence embedding," in *Proceedings of the eleventh ACM International Conference on Web Search and Data Mining*, pp. 565–573, Marina Del Rey CA USA, 2018.
- [8] B. Hidasi and A. Karatzoglou, "Recurrent neural networks with top-k gains for session-based recommendations," in *Proceedings of the 27th ACM international conference on information and knowledge management*, pp. 843–852, Torino Italy, 2018.
- [9] W. C. Kang and J. McAuley, "Self-attentive sequential recommendation," in *2018 IEEE International Conference on Data Mining (ICDM)*, pp. 197–206, Singapore, 2018.
- [10] S. Sang, N. Liu, W. Li, Z. Zhang, Q. Qin, and W. Yuan, "High-order attentive graph neural network for session-based recommendation," *Applied Intelligence*, pp. 1–15, 2022.
- [11] J. Zhang, X. Mu, P. Zhao, K. Kang, and C. Ma, "Improving current interest with item and review sequential patterns for sequential recommendation," *Engineering Applications of Artificial Intelligence*, vol. 104, article 104348, 2021.
- [12] T. Bai, P. Du, W. X. Zhao, J. R. Wen, and J. Y. Nie, "A long-short demands-aware model for next-item recommendation," 2019, <http://arxiv.org/abs/1903.00066>.
- [13] C. Xu, J. Feng, P. Zhao et al., "Long- and short-term self-attention network for sequential recommendation," *Neuro-computing*, vol. 423, pp. 580–589, 2021.
- [14] Z. Pan, F. Cai, W. Chen, C. Chen, and H. Chen, "Collaborative graph learning for session-based recommendation," *ACM Transactions on Information Systems*, vol. 40, no. 4, pp. 1–26, 2022.
- [15] F. Lv, T. Jin, C. Yu et al., "SDM: sequential deep matching model for online large-scale recommender system," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 2635–2643, Beijing China, 2019.
- [16] J. Tang, F. Belletti, S. Jain et al., "Towards neural mixture recommender for long range dependent user sequences," in *The World Wide Web Conference*, pp. 1782–1793, San Francisco CA USA, 2019.
- [17] J. Sun, Y. Zhang, W. Guo et al., "Neighbor interaction aware graph convolution networks for recommendation," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1289–1298, Virtual Event China, 2020.
- [18] Z. Li and L. Zhang, "Fast neighbor user searching for neighborhood-based collaborative filtering with hybrid user similarity measures," *Soft Computing*, vol. 25, no. 7, pp. 5323–5338, 2021.
- [19] S. Banerjee, P. Banjare, B. Pal, and M. Jenamani, "A multistep priority-based ranking for top-N recommendation using social and tag information," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 2, pp. 2509–2525, 2021.
- [20] Y. Guo, Y. Ling, and H. Chen, "A time-aware graph neural network for session-based recommendation," *IEEE Access*, vol. 8, pp. 167371–167382, 2020.
- [21] Y. Ding and X. Li, "Time weight collaborative filtering," in *Proceedings of the 14th ACM international conference on Information and knowledge management*, pp. 485–492, Bremen Germany, 2005.
- [22] N. Lathia, S. Hailes, and L. Capra, "Temporal collaborative filtering with adaptive neighbourhoods," in *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pp. 796–797, Boston MA USA, 2009.
- [23] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized Markov chains for next-basket recommendation," in *Proceedings of the 19th international conference on World wide web*, pp. 811–820, Raleigh North Carolina USA, 2010.
- [24] R. He and J. McAuley, "Fusing similarity models with Markov chains for sparse sequential recommendation," in *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pp. 191–200, Barcelona, Spain, 2016.
- [25] S. Kabbur, X. Ning, and G. Karypis, "Fism: factored item similarity models for top-n recommender systems," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 659–667, Chicago Illinois USA, 2013.
- [26] N. Sahoo, P. V. Singh, and T. Mukhopadhyay, "A hidden Markov model for collaborative filtering," *MIS Quarterly*, vol. 36, no. 4, pp. 1329–1356, 2012.

- [27] C. Lonjarret, R. Auburtin, C. Robardet, and M. Plantevit, "Sequential recommendation with metric models based on frequent sequences," *Data Mining and Knowledge Discovery*, vol. 35, no. 3, pp. 1087–1133, 2021.
- [28] L. Huang, Y. Ma, S. Wang, and Y. Liu, "An attention-based spatiotemporal lstm network for next poi recommendation," *IEEE Transactions on Services Computing*, vol. 14, no. 6, pp. 1585–1597, 2019.
- [29] M. Gan and K. Xiao, "R-RNN: extracting user recent behavior sequence for click-through rate prediction," *IEEE Access*, vol. 7, pp. 111767–111777, 2019.
- [30] H. Ying, F. Zhuang, F. Zhang et al., "Sequential recommender system based on hierarchical attention network," in *IJCAI International Joint Conference on Artificial Intelligence*, Sweden, Stockholm, 2018.
- [31] J. Lin, W. Pan, and Z. Ming, "FISSA: fusing item similarity models with self-attention networks for sequential recommendation," in *Fourteenth ACM Conference on Recommender Systems*, pp. 130–139, Virtual Event Brazil, 2020.
- [32] M. Quadrana, A. Karatzoglou, B. Hidasi, and P. Cremonesi, "Personalizing session-based recommendations with hierarchical recurrent neural networks," in *Proceedings of the Eleventh ACM Conference on Recommender Systems*, pp. 130–137, Como Italy, 2017.
- [33] Y. Feng, B. Zhang, B. H. Qiang, Y. Y. Zhang, and J. X. Shang, "MN-HDRM: a novel hybrid dynamic recommendation model based on long-short-term interests multiple neural networks," *Chinese Journal of Computers*, vol. 42, no. 1, pp. 16–28, 2019.
- [34] K. Hu, L. Qiu, S. Zhang, Z. Wang, and N. Fang, "An incremental rare association rule mining approach with a life cycle tree structure considering time-sensitive data," *Applied Intelligence*, pp. 1–25, 2022.
- [35] H. Wang, P. Li, Y. Liu, and J. Shao, "Towards real-time demand-aware sequential POI recommendation," *Information Sciences*, vol. 547, pp. 482–497, 2021.
- [36] C. Ma, L. Ma, Y. Zhang, J. Sun, X. Liu, and M. Coates, "Memory augmented graph neural networks for sequential recommendation," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 4, pp. 5045–5052, 2020.
- [37] A. Mirzal, "Statistical analysis of microarray data clustering using NMF, spectral clustering, Kmeans, and GMM," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 19, pp. 1173–1192, 2020.
- [38] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," 2015, <http://arxiv.org/abs/1511.06939>.