IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 22 September 2023, accepted 4 October 2023, date of publication 18 October 2023, date of current version 26 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3325476

RESEARCH ARTICLE

A Graph Neural Network Recommendation Method Integrating Multi Head Attention Mechanism and Improved Gated Recurrent Unit Algorithm

FANG LIU, JUAN WANG, AND JUNYE YANG
College of Internet Application Technology, Shijiazhuang Institute of Technology, Shijiazhuang 050228, China
Corresponding author: Junye Yang (yjy20230315@163.com)

ABSTRACT To improve the accuracy of graph neural network recommendation algorithms, research mainly integrates multi head attention mechanism and GRU, which is to construct a graph neural network recommendation model; Considering the long and short term preferences of users, a graph neural network algorithm integrating long and short term preferences is constructed. The research results indicated that when the embedding dimension was 64, the batch size of selected samples was 64, the learning rate was 0.0005, the vertical stacking layer of GRU was 2, the iteration period was 5, and the dropout probability was 50% with the best performance. The graph neural network algorithm based on long and short term preferences had higher recommendation accuracy compared to other algorithms. Modeling users' short-term and long-term preferences can achieve the goal of comprehensively improving recommendation effectiveness.

INDEX TERMS Graph neural networks, attention mechanism, short term preference, recommendation systems, feature extraction.

I. INTRODUCTION

On the Internet, the rapid growth of information has brought about the information overload. Recommendation systems have gained significant attention as they prove to be an effective solution to this problem [1]. In recent years, recommendation systems have made significant progress, and their research has become a hot topic of concern from all walks of life. Graph neural network (GNN) has been widely used in recommendation systems due to their advantages in graph representation learning [2]. However, due to the fact that most of the data in recommendation systems is graph structured, introducing graph neural networks into recommendation systems can better extract user preferences and needs from massive data. Therefore, using graph neural networks for recommendation is currently a very important research direction in recommendation systems. In graph neural network recommendations, users' interests and preferences are influenced by multiple factors such as their historical behavior and social network, and exhibit a certain degree of dynamism [3]. How to integrate users' social network information and temporal interests to extract useful information from recommendations is currently a challenge faced by recommendation systems. This study has five parts. The first part provides a brief introduction to the research background. The second part is a literature review that summarizes the research on recommendation systems by scholars in different fields and introduces the purpose of this study. The third part is the research method, which constructs a GNN algorithm that integrates multi head attention mechanism and GRU, and a graph neural network algorithm that integrates long and short term preferences; The fourth part is the result analysis, mainly focusing on the evaluation and performance analysis of research methods; The final part is the conclusion, including the main conclusions and prospects of this study.

The associate editor coordinating the review of this manuscript and approving it for publication was Jon Atli Benediktsson.

II. RELATED WORKS

Introducing graph neural networks into recommendation systems can better mine users' interests and needs from massive data. Therefore, using graph neural networks for recommendation is currently a very important research direction in recommendation systems, and scholars in different fields have conducted extensive research.

Du S et al. proposed a image recommendation algorithm to address the issue of low efficiency in traditional social network image recommendation algorithms. The algorithm sorted user interaction records over time, combined feature algorithms to construct feature vectors, and constructed a long and short term memory (LSTM) neural network. Results showed that LSTM's performance was superior to traditional social network image recommendation algorithms [4]. To enhance the recommendation accuracy, scholars like Fang et al. proposed a Collaborative Filtering recommendation algorithm that leverages deep neural network fusion. This method combined short-term memory networks and deep neural networks to extract additional attributes, effectively reducing the root mean square error and mean absolute error, which were 2.015% and 2.222% lower than the original method, respectively [5]. Guo and Wang proposed a social recommendation framework using deep neural networks to address the issue of ignoring the correlation between project features in current IoT user recommendation systems, which abstractly encoded the user and item feature space, considered the correlation between IoT user characteristics, and provided more feasible personalized information services [6]. Zhao et al. and other scholars have designed a hierarchical attention recommendation system using social networks to address the neglect of heterogeneity in current graph neural network recommendation algorithms. This system utilized user behavior information from social networks and integrated information from heterogeneous human networks. The research results indicated that this system can effectively improve recommendation systems and networks flexibility [7]. Zhu et al. designed a neural attention travel package recommendation system based on long and short term behavior to address current travel package recommendations not effectively meeting user preferences. The system utilized travel package encoders and user encoders to fuse users' long and short term preferences. The experimental results showed that the system can extract user preferences and obtain tourism package recommendation results that better meet user preferences [8].

Huang L et al. proposed an interactive recommendation system using deep reinforcement learning to address long-term recommendation functionality shortage in recommendation systems. The model was optimized using reinforcement learning. The experimental results indicated that the long-term recommendation hit rate of the research system was superior to commonly used methods [9]. To enhance the efficiency and quality of human resource recommendation systems, Zhu H developed a machine learning-based system for human resource recommendations. The system utilized the gradient descent algorithm to enhance the prediction accuracy and was trained using a backpropagation neural network. The experimental results indicated that the system can improve the accuracy of human resource recommendations [10]. To enhance online teaching resources recommendation, Yuan Q constructed a network education resource recommendation model based on path sorting method. Using path sorting algorithm to predict and sort links in the learning library, the research results showed that the research method can enhance online teaching resources recommendation and teaching resources sharing [11]. Liu Z and other scholars designed a personalized label recommendation algorithm based on convolutional features and weighted random walks to enhance social image recommendation, and analyzed the correlation of user images. The experimental results indicated that the research method can effectively portray semantic information present in images and cater to individual user preferences [12]. Chen M and other researchers have designed a hybrid service combination recommendation method to address the difficulty of creating suitable service processes based on current users' needs, taking into account factors such as user interests. The research results indicated that it can improve recommending relevant service components to users [13].

The above research showed that a graph neural network algorithm that integrated attention mechanisms and short-term preferences can enhance recommendation systems. Given the limited research on integrating both in graph neural network recommendation algorithms, this study will integrate attention mechanisms and short-term preferences, combined with feature extraction for the performance enhancement of the recommendation algorithm.

III. GRAPH NEURAL NETWORK ALGORITHM INTEGRATING MULTI HEAD ATTENTION MECHANISM AND LONG SHORT TERM PREFERENCE

This chapter first constructed a graph neural network model, and on this basis, constructed a GNN algorithm that integrates multi head attention mechanism and GRU. To consider long and short term preferences of users, a recommendation method for a graph neural network that combines long and short preferences is proposed.

A. GNN MODEL CONSTRUCTION

GNN is a new method inspired by Convolutional Neural Networks (CNN), which can utilize the node information in the network and the topological structure of the graph for learning and effectively fuse multi hop association information. In graph neural networks, aggregation and update modules are generally used to embed nodes into the network [14]. This algorithm utilizes an aggregation module to aggregate neighboring nodes and update them. On this basis, combined with the weights of each edge between each node, the number of transfers from each node to its adjacent nodes is determined, and the weights between each node are determined. A graph is the most commonly used data structure that can model a series of nodes or edges. In recommendation systems, most data is a graph structure, where users and projects can be represented by nodes, while edges represent the correlations between nodes. However, traditional deep learning methods cannot fully utilize multi-source data, which can easily lead to information loss and affect the performance of recommendation systems. In recent years, the rapid emergence of massive graph data such as social networks, molecular structures, and knowledge graphs has brought new opportunities for the research of GNN. Its emergence derives from the development of CNN and graph representation learning. The general framework of a neural network recommendation system is shown in Figure 1.

In Figure 1, first, Data modeling is modeled as a graph structure, and then, according to the embedded relationship between users and projects, data is analyzed using graph neural network. On this basis, this method is used to predict the potential interests of each project and generate corresponding recommendation lists accordingly. For various studies, the graph theory models and algorithms used by people also vary. CNN has a good application prospect in extracting the features of traditional Euclid local data such as images and characters. However, because a large number of applications produce a large number of non Euclid data, CNN must have a certain generalization ability to adapt to the uncertainty of the size of the operation target. In graph representation learning (GRL), the main focus is on how to generate low dimensional vectors for nodes, edges, and subgraphs, and use them to express the complex structure of the graph [15]. To solve this problem, a graph neural network was formed by combining CNN and GRL. GNN have strong representation capabilities and can achieve high-quality embedding.

In addition, the correctness of recommendations depends on the similarity between the obtained users and the project, which should be reflected in the embedding process. Meanwhile, due to the existence of certain common preferences among users, the interaction between them becomes closer, resulting in collaborative filtering effects and improving the accuracy of recommendations. Traditional recommendation systems only provide direct interactive recommendations to users, ignoring higher-level connectivity, resulting in a decrease in recommendation effectiveness. In stark contrast, recommendation systems using GNN have become the latest research direction in recommendation systems due to their ability to process structural data and mine structural information, capturing high-order connectivity. Generally speaking, graph neural network models can be subdivided into spectral and spatial models. The spectral model treats images as signals and processes them using graph convolution. This method first performs Fourier transform on the graph signal into the spectral domain, and then processes it through a filter before converting it into a spatial domain signal. The expression for processing signals using filters is shown in equation (1).

$$g \circ x = F^{-1} \left(F(g) \otimes F(x) \right) \tag{1}$$

In equation (1), g and x represent the filter and signal, respectively, and F represents the Fourier transform [16]. Another spatial model is directly convolutional, and the features of the signal part in the local image are extracted using weighted aggregation method. Although the starting points of the two models are different, their basic principle is to continuously collect information from neighboring nodes in the network, thereby obtaining high-order connectivity. Here, information is called embedding, which is a low dimensional vector. To achieve this, the graph neural network needs to first embed the domain, then aggregate the target nodes or edges, and then update them layer by layer.

B. GRAPH NEURAL NETWORK ALGORITHM INTEGRATING MULTI HEAD ATTENTION MECHANISM AND GATED RECURRENT UNIT

As information technology continues to evolve and the Internet becomes more widespread, online interaction has penetrated into people's daily lives, providing new ideas for the development of recommendation systems. In this context, people have begun to introduce attention mechanisms into recommendation systems and have demonstrated through experiments that they can effectively improve the recommendations. Treisman et al. first proposed an attention mechanism, which essentially captured information based on the probability distribution of attention. Over the past few years, attention mechanisms have found extensive applications in natural language processing, computer vision, and image recognition. Since 2017, attention has also been introduced into recommendation systems. In a recommendation system, users' interests and preferences are influenced by factors such as their historical behavior and social network, showing dynamic changes. However, how to integrate users' social networks and time interests and extract useful information from recommendations is currently a challenge faced by recommendation systems.

Assuming $\{U \left| u_1, u_2, \ldots, u_n \right.\}$ and $\{V \left| v_1, v_2, \ldots, v_n \right.\}$ are used to describe the user set and item set respectively, then the vertex set of the graph is $U \cup V$. Among them, n and m represent the total number of users and items, respectively [17]. In the graphs network, there are two types of edges, one is the interaction relationship between users and objects, as shown in equation (2).

$$\left\{ R \left| U^{*} V \right. \right\} = \left\{ R \left| r_{i,j} = \left(u_i, v_j \right), u_i \in U, v_j \in V \right. \right\} \tag{2}$$

In formula (2), U and U represent the set of users and the set of items, respectively, while R represents the interaction relationship between users and items.Equation (2) describes the evaluation of item v_j by user Au_i, with its weight denoted as $w^R_{i,j}$, which represents the rating given by user u_i to item v_j. The timestamp of each user and item is denoted as t, which represents the time when the rating occurred. The second type refers to an undirected interaction between users, as expressed in equation (3).

$$\left\{ S \left| U^{*} U \right. \right\} = \left\{ S \left| s_{i,j} = \left(u_i, v_j \right), u_i \in U, v_j \in U \right. \right\} \tag{3}$$

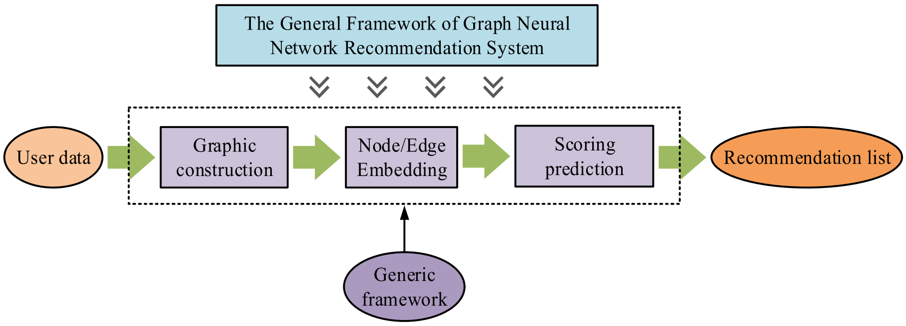


FIGURE 1. General framework of graph neural network recommendation system.

Equation (3) refers to user u_i paying attention to user u_j in social networks. Therefore, the weight of this edge is defined as $p(i, j)$, which refers to the social connection between user u_i and u_j. Therefore, in this study, recommendation systems can be described as user set U, item set V, user model, and social model. By training the algorithm to obtain a prediction function, it can predict the score $w^R_{i,j}$ of user u_i on item v_j, and use it as a basis for recommendation. A GNN (MGRU) algorithm that integrates multi head attention mechanism (MHAM) and Gated Recurrent Units(GRU) has been developed. Firstly, a gated recurrent unit is used to store and forget temporal information, improving the abstraction of node iteration in the graph neural network. Then, different influences of friends on users are obtained through attention memory networks, and the influence of friends is regulated through multi head attention mechanisms. On this basis, a gated neural network is used to fuse the influence of friends and user preferences to achieve project recommendation. The graph network recommendation algorithm framework based on MGRU is shown in Figure 2.

From Figure 2, it can be seen that the MGRU based graph network recommendation algorithm framework mainly includes the user item model section and the user social model section. Among them, the former mainly calculates the set of nodes near the user and sorts the items according to time series. It mainly uses GRU to memorize and forget temporal information, enhancing the abstract ability of graph neural networks during node iteration. The latter obtains different influences of friends on users by paying attention to the attention memory network, and regulates the influence of friends. Finally, the calculation results of the two modules are synthesized through a gated network, and the final predicted component is obtained through matrix decomposition. The user item model is a graph based on user node u_i and scoring edge R_{u_i} starting from that user node. In the figure, the area where each user is located is a project that has already been evaluated by that user. The collection of these projects is named $N^V_{(ui)}$ and then sorted in chronological order t. In order to extract time series features of user rated items, the study will use GRU to extract their rating sequence features in the user item model.

Embedding represents the distribution among all user sets, calculating the adjacent node set $N^V_{(ui)} = Neigh^V_{(ui)}$ of user u_i, and recording the number of nodes as L; The items in the $N^V_{(ui)}$ set are arranged in ascending order according to time t to obtain the item sequence $S(u_i) = [v_1, v_2, \ldots, v_L]$. The embedding vector e^v_j is queried corresponding to all items v_j in $S(u_i)$, and if the scores are all $w^R_{i,j}$, then its embedding vector is $e^R_{i,j}$. e^v_j and $e^R_{i,j}$ are spliced into $x_j = CONCAT\left(e^v_j, e^R_{i,j}\right)$ and input into the perceptron for dimensionality reduction, as shown in equation (4) [18].

$$e'_j = \mathrm{Re}LU\left(Wx_j + b\right) \tag{4}$$

In equation (4), ReLU represents the activation function, and e'_j is used to describe the item vector containing the score, and e'_j is used to update the item sequence S. By inputting the updated S and state h simultaneously, equation (5) can be obtained.

$$S' = \left[h^{(k)}_1, h^{(k)}_2, \ldots, h^{(k)}_L\right] \tag{5}$$

In equation (5), the number of stacking layers is $k = 2$, and GRU is calculated at time t. By concatenating the input x_t of the current node with the state $h_{(t-1)}$ of the previous time, combined with the fully connected layer, it is possible to obtain r_t control reset gating and z_t control forgetting gating. The range of values for r_t and z_t is, and the closer the value of z_t is to 1, the more data is memorized; On the contrary, the closer the z_t value is to 0, the more data is forgotten. After obtaining the gating signal, it is possible to use r_t to reset the state of the previous moment, then concatenate the results with the input, adjust the data value range to between $[-1, 1]$, and obtain n_t. Selective forgetting and memory of the information in n_t completes that of input and temporal information. Using state $h^{(k)}_L$ as a reference, weighted average the L elements in S to obtain h', concatenate it with $h^{(k)}_L$, and activate it through the ReLU function to obtain y_i. Then the attention mechanism is realized by a single output based perceptron to obtain the weight coefficient p_i, and finally the aggregation result u^{agg}_i is obtained. The embedding vector e^u_i and aggregation result u^{agg}_i of the node itself are spliced and activated to finally obtain the representation $u^{(n)}_i$ of the node after n iterations, as shown in equation (6).

$$\begin{cases} x_i = CONCAT\left(u^{agg}_i, e^u_i\right) \\ u^{(n)}_i = \mathrm{Re}LU\left(W_{xi} + b\right) \end{cases} \tag{6}$$

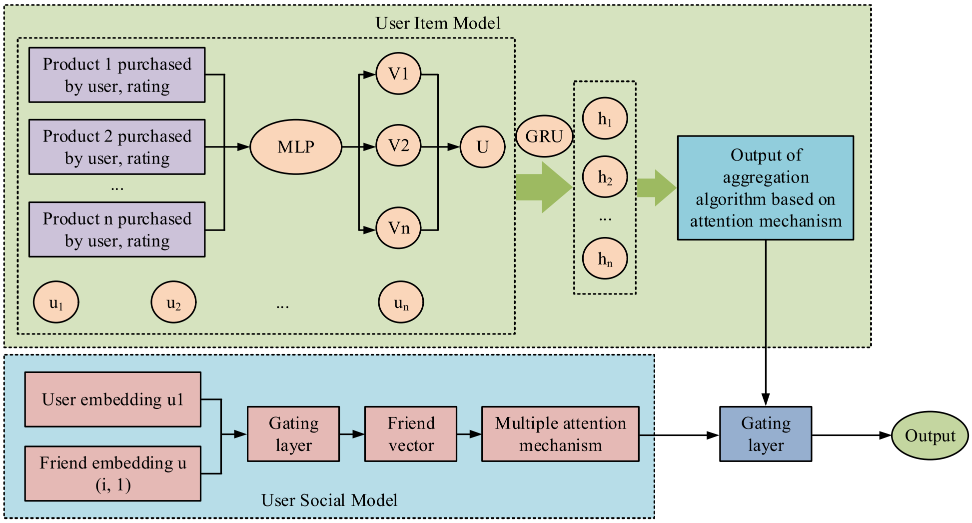


FIGURE 2. MGRU based graph network recommendation algorithm framework.

The user social model is a local graph composed of user node u_i and friend node u_j. In this algorithm, edges are undirected, and each edge represents the social connection between users. Neighborhood nodes are a collection of users who care about each other, that is, users and their friend pairs. For all users u_i and user friend $u_{(i,j)}$, the embedding vector is obtained by combining the gated network, and the expression for the embedding representation is shown in equation (7).

$$\begin{cases} S_{gate} = \tanh\left(w_1 * u_i + w_2 * u_{(i,j)} + b\right) \\ S = S_{gate} * u_i + \left(1 - S_{gate}\right) * u_{(i,j)} \end{cases} \tag{7}$$

The social relationship vector between users and friends can be learned through the attention memory module, where Q is the memory matrix, Q_l is each memory segment, and d is the embedding dimension. The embedded expression of a user and a friend is input, and then the common interest preference vector of the friend and the user is output. The commonly used attention mechanism is to balance friends influence, and the weight results only change when users interact with each other. But in reality, users do not treat every friend's opinion equally. Friends who have deep knowledge in a certain aspect often give more accurate opinions. The multi head attention mechanism can extract multiple features and also focus on providing suggestions to friends in certain aspects, thus providing users with more accurate information they want. The MHAM structure is shown below.

From Figure 3, MHAM combines friend embedding and user embedding, and after scaling dot product attention and connection processing, the friend influence vector is obtained. Therefore, this study will choose MHAM to obtain the key content that users pay attention to in social information, to obtain the importance of friends to users in different aspects. The multi head attention mechanism utilizes multiple linear mappings of variables Q, K, and V, and utilizes the dot product attention operation to concatenate the scaled dot product attention results into the linear mapping, thereby obtaining the solution results of the algorithm. Each user Q

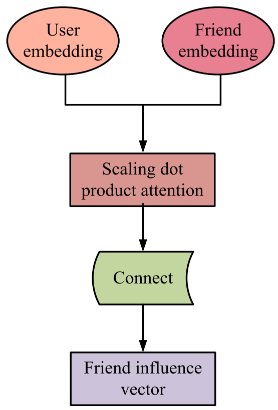


FIGURE 3. The MHAM structure.

is compared with their friend vector K to obtain the output of each single head, as shown in equation (8).

$$head_j = Attention\left(Q, K, V\right) = soft\max\left\{\frac{Q \cdot K^T}{\sqrt{d_Q}}\right\} V \tag{8}$$

In equation (8), $\sqrt{d_Q}$ represents the characteristic dimensions of vectors Q and K, which has the effect of avoiding excessive inner product values. Meanwhile, K and V are the same variable $f_{(i,j)}$F and then the results of the single head are concatenated and linearly transformed to obtain the final result, as shown in equation (9) [19].

$$\begin{aligned} p_{(i,j)} &= MultiHeadAttention\left(Q, K, V\right) \\ &= Concat\left(head_1, head_2, \ldots, head_j\right) \end{aligned} \tag{9}$$

In equation (9), all weights $p_{(i,j)}$ are fused with embedded $f_{(i,j)}$. By combining the obtained friend influence vector, the user's comprehensive interest preference can be obtained.Using the friend influence vectors obtained from the above two modules, fuse this vector with the user's own interests to obtain the user's comprehensive interest preferences. Affected by the gating mechanism in long-term and short-term memory networks, gating G and the resulting user's comprehensive interest I are influenced by different aspects of different friends. Integrate the friend vector with the user itself, and use the gating mechanism in the memory network to obtain the user's interest preference I_i.

C. GRAPH NEURAL NETWORK ALGORITHM INTEGRATING LONG AND SHORT TERM PREFERENCES

Currently, recommendation algorithms based on graph neural networks generally obtain users' long-term preferences using their entire historical interaction content, while ignoring the fact that users' short-term preferences change over time. Therefore, based on the fusion of multi head attention mechanism and GRU graph neural network algorithms, a recommendation method combining long and short preferences in graph neural networks is proposed.

In recent years, domestic and foreign researchers have established a graph model using the relationship between users and projects in the network, and aggregated and updated its nodes to achieve embedded representation of users and projects. However, existing graph neural network modeling methods often only analyze one aspect of the long and short term preferences (LSTP) generated by users. The essence of user preferences is a combination of LSTP. Long term preferences record a user's long-term habits, while short-term preferences are new interests that arise over time. To this end, a Combined Graph Neural Network With Long-term and Short term Recommendation (GNNLSR) algorithm is proposed based on LSTP. At the same time, by fusing the extraction of items, it can not only capture users' LSTP, but also combine them with project features to achieve better recommendation results.

Some items in the sequence of user u_i and item v_j constitute $S^{ui} = \left(S_1^{ui}, S_2^{ui}, \ldots, S_t^{ui}\right)$. In $S^{ui} = \left(S_1^{ui}, S_2^{ui}, \ldots, S_t^{ui}\right)$, index t is used to describe the order in which item interactions occur in sequence S^{ui}. The interaction between user u_i and item v_j is represented by edge $r_{i,j}$, with edge weight $w_{i,j}^R$, which is used to describe user u_i's rating of item v_j. GNNLSR includes the user model and the item model. The structure of the user model is shown in Figure 4.

In Figure 4, in terms of short-term preference model, the items that interact with short-term preferences based on the interaction between users are first associated with scoring, and then a GNN is used to model them. In the long-term preference model, users who are similar to them are found, embedded information is added, and then summarized to obtain their long-term preferences. Firstly, after passing through the entire set of users and their most recent interacting t item sequence, the embedded representation for user u_i's interaction is obtained, as shown in equation (10).

$$T\left(u_i\right) = \left[e_1^v, e_2^v, \ldots, e_t^v\right] \tag{10}$$

Due to the varying impact of all items that users interact with in the short term, a rating vector should be embedded. The embedding vector $e_{i,t}^R$ of the score $w_{i,t}^R$ corresponding to all items v_t is queried, e_t^v and $e_{i,t}^R$ are spliced $x_t = CONCAT\left(e_t^v, e_{i,t}^R\right)$, and the dimension reduction of perception machine is input, as shown in equation (11).

$$e_t' = \mathrm{Re}LU\left(Wx_t + b\right) \tag{11}$$

In equation (11), ReLU represents the activation function, and e_t' describes the item vector involving scoring. e_t' is used to update the item sequence T to obtain the embedded item representation $T\left(u_i\right) = \left[e_1^{v'}, e_2^{v'}, \ldots e_t^{v'}\right]$ containing scoring. Next, each user u_i and item sequence $T\left(u_i\right)$ are embedded in interaction graph into the vector to obtain the node representation of the user short-term preference model through the transfer and aggregation of neighboring nodes [20]. After n operations, the user's n node representations can be obtained, and then the n node levels are combined to obtain the short-term preferences in equation (12).

$$u_i^{(n)} = u^{(0)} \left|\left|u^{(1)}\right|\right| \cdots \left\|u^{(n)}\right. \tag{12}$$

However, existing graph neural network recommendation models mainly focus on users' short-term interactions, while neglecting their long-term preferences. They only use users' embedded representations to model their long-term preferences, which is not accurate enough. In the research, the first step is to find similar users, that is, two users who are interacting with the same project. This step can improve the issue of cold start by providing more information to a few users in the project evaluation. Enter a user u_i and their interaction sequence S^{u_i}. Among them, each item in the interaction sequence $\left(S_{t-L}^{u_i}, S_{t-L+1}^{u_i}, \ldots, S_{t-1}^{u_i}\right)$ is arranged in the order in which it interacts with all users, as shown in Figure 5.

From Figure 5, it can be seen that if user u_{i-1} is found in the list to have the same rating as current user u_i and interaction project v_j, then user u_i's similar user regarding project v_j is user u_{i-1}. According to this method, similar users corresponding to all users u_i and all projects can be found. Afterwards, Dior's similar user embeddings and supplementary features C_u are aggregated. Assuming that user u_i has 5 interaction items, the corresponding similar users can be found through the interaction items. Combining the user's embeddings, the long-term preference u_i' is represented. Attention mechanism is used to capture associations between users and merge their characteristics to obtain supplementary feature P_u, as shown in equation (13).

$$P_u = \sum_{k=1}^{K} Z_{S_i} u_i \times u_{S_k^{u_i}} \tag{13}$$

In equation (13), $Z_{S_i} u_i$ represents the weight between users. Finally, user features and supplementary features are merged to obtain the final result, as shown in equation (14).

$$u_i' = CONCAT\left(u_i, p_u\right) \tag{14}$$

In the item model, this module mainly extracts the interactive information and corresponding scores, and obtains the potential feature $\mathrm{A}v_j'$ of the item. The item model structure is shown below.

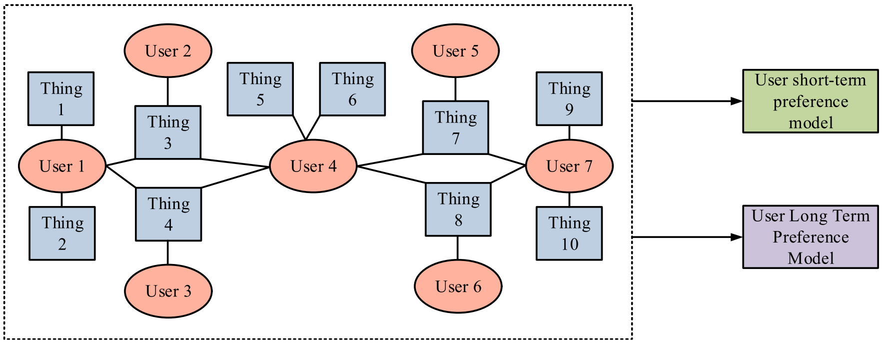


FIGURE 4. The structure of the user model.

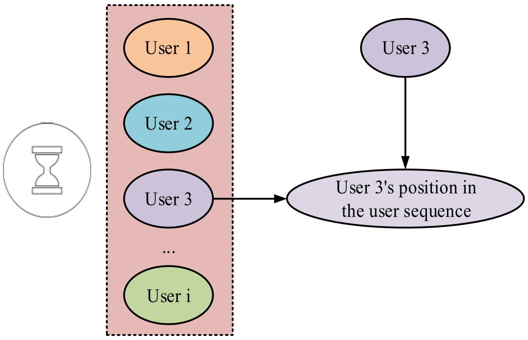


FIGURE 5. Similar user search process.

Its goal is to learn the potential characteristics of items from the items, including the interactive information between items and the corresponding score. The relationship between them is the interactive information, and the level of user evaluation is the user's item satisfaction. Therefore, fully exploring the potential features contained in both is important for recommendation systems.

IV. EVALUATION AND PERFORMANCE ANALYSIS OF GRAPH NEURAL NETWORK ALGORITHM

This chapter evaluates and analyzes the performance of GNN that integrates MHAM and GRU, as well as a graph neural network algorithm that integrates LSTP. Using a multi head attention machine can more fully consider the various impacts of friends on users, highlighting the importance of friends in a certain field in depth. Finally, employing a gated network to fuse the model has been shown to enhance the accuracy of recommendations. LSTM fusion method using GNN excavates users' short-term preferences by analyzing their recent interaction information, and obtains stable long-term preferences through relationships with similar users. By combining them with project features, it effectively improves recommendation results.

A. EVALUATION AND PERFORMANCE ANALYSIS OF GRAPH NEURAL NETWORK ALGORITHM INTEGRATING MULTI HEAD ATTENTION MECHANISM AND GATED RECURRENT UNIT

In this study, two representative datasets, Ciao and Epions, were used to test the effectiveness of recommendation algorithms, including user rating information (0-5 points), timestamps, and indexes of users and followers, and these data were statistically analyzed. The Ciao dataset is a user evaluation dataset from the Ciao e-commerce website. This dataset contains user ratings and comments on various products. Through these ratings and comments, a user product rating matrix can be constructed for the research and evaluation of recommendation systems. The Ciao dataset is typically used to evaluate the personalized recommendation ability and accuracy of recommendation algorithms. The Epinions dataset is a user opinion dataset from the Epinions website. This dataset contains users' opinions, ratings, and comments on different products or services, such as books, movies, electronic products, etc. These opinions and rating information can be used to construct a user product rating matrix for research and evaluation of recommendation systems. The Epions dataset is often used to evaluate the effectiveness and scalability of recommendation algorithms in real-world scenarios. Ciao and Epions are two representative datasets for recommendation systems in the field of e-commerce. These datasets are commonly used standard datasets in the field of recommendation algorithm research, and due to their authenticity and diversity, they can provide beneficial verification of the performance and applicability of recommendation algorithms. By conducting experiments and evaluations on these datasets, researchers can better understand the performance of recommendation algorithms in real scenarios and make targeted improvements and optimizations.Due to MGRU being a recommendation algorithm that combines temporal and social information, this study aims to exclude user data without social information, retaining only user data with one social item, and removing some duplicate items with scores below 4. Considering that users can estimate the score of items that have not been rated, the research used the root mean square error (RMSE) and mean absolute error (MAE) commonly used in the recommendation system as indicators of the accuracy of prediction scores. The study selected 70% of the Ciao and Epionions datasets as the training set, 20% of the sample set as the validation set, and the remaining sample set as the test sample set. On this basis, the study analyzed the impact of experimental embedding dimension, batch size of selected samples, learning rate, number of iteration cycles, GRU layers, and Dropout probability on

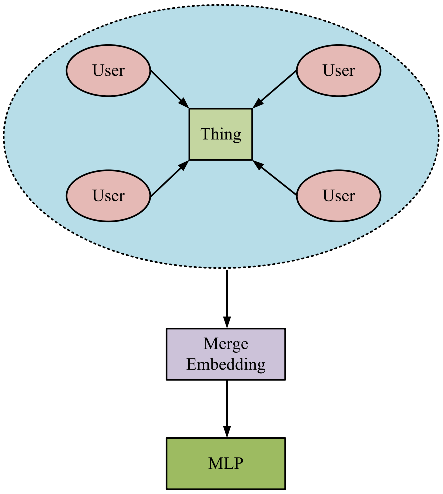


FIGURE 6. Structure of the item model.

the recommendation performance of MGRU algorithm. After obtaining the optimal parameters, the following sections were compared with other recommendation algorithms. Results of embedding dimensions and batch values were obtained, as shown in Figure 7.

Figures 7 (a) and 7 (b) respectively represent the impact of embedded samples on MAE and RMAE; Figures 7 (c) and 7 (d) respectively represent the impact of batch values on MAE and RMSE. From the graph, it can be seen that as the dimension increased, the values of MAE and RMSE decreased until the embedding dimension was 64, and the performance started to decline again. This means that when the embedding dimension is 64, the algorithm can more accurately predict the target value, which is closer to the actual observation value. This phenomenon may be due to the 64 dimensional embedding space being sufficiently expressive to effectively capture features and complex relationships in the data, while also possessing a certain degree of generalization ability, avoiding overfitting and underfitting problems.This indicated that the size of the embedded dimension can to some extent affect the results of the algorithm. The size of the algorithm for training a set of data was determined by the batch values. When the values were too small, the training time became longer due to excessive segmentation of the data, and the data was also difficult to converge; If the numerical value was too large, it not only shortened the training time, but also degraded the performance. Therefore, when the value of MGRU was 64, its performance was the best. Afterwards, the MGRU algorithm was analyzed based on the learning rate and GRU layers obtained from experiments using the Ciao and Epions datasets, as shown in Figure 8.

Figures 8 (a) and 7 (b) respectively show the impact of learning rate on MAE and RMAE; Figures 7 (c) and 7 (d) respectively show the impact of GRU layers on MAE and RMSE. From the graph, the best learning rate for MGRU was 0.0005. When the learning rate was lower than 0.0005, the performance of the model was not very poor, but it did not achieve the best performance and led to a long training time for the model; However, when the learning rate was greater than 0.001, the effectiveness of this method significantly decreased, and at the same time, an excessively high learning rate indeed caused the model to lose its optimal parameters, thereby reducing its performance. This means that a learning rate of 0.0005 can effectively update model parameters in a reasonable time, enabling the model to better fit training data and accurately predict target values, and

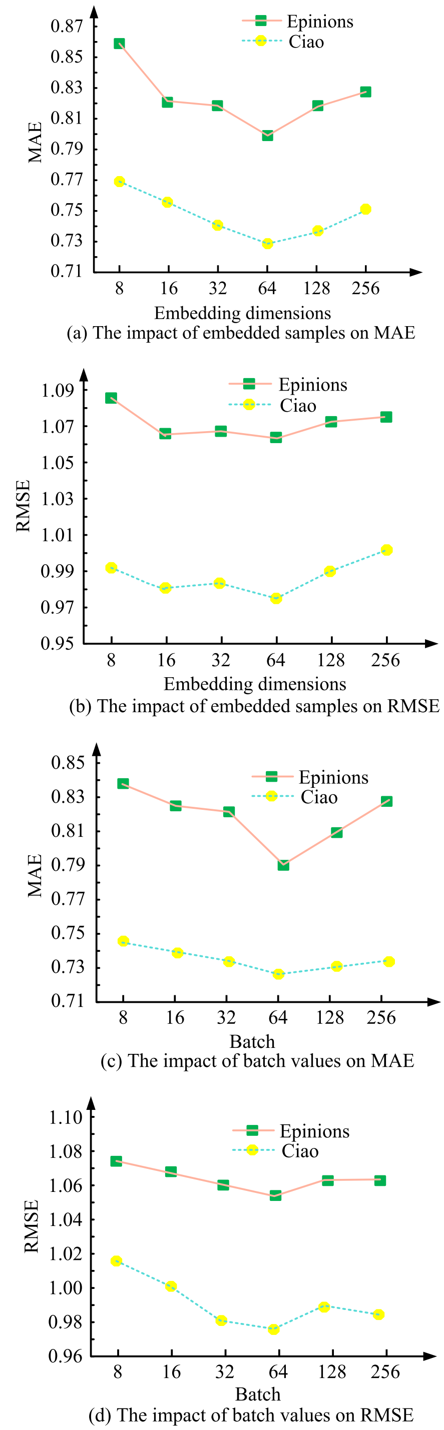


FIGURE 7. The influence of embedded samples and batch values on MAE and RMSE.

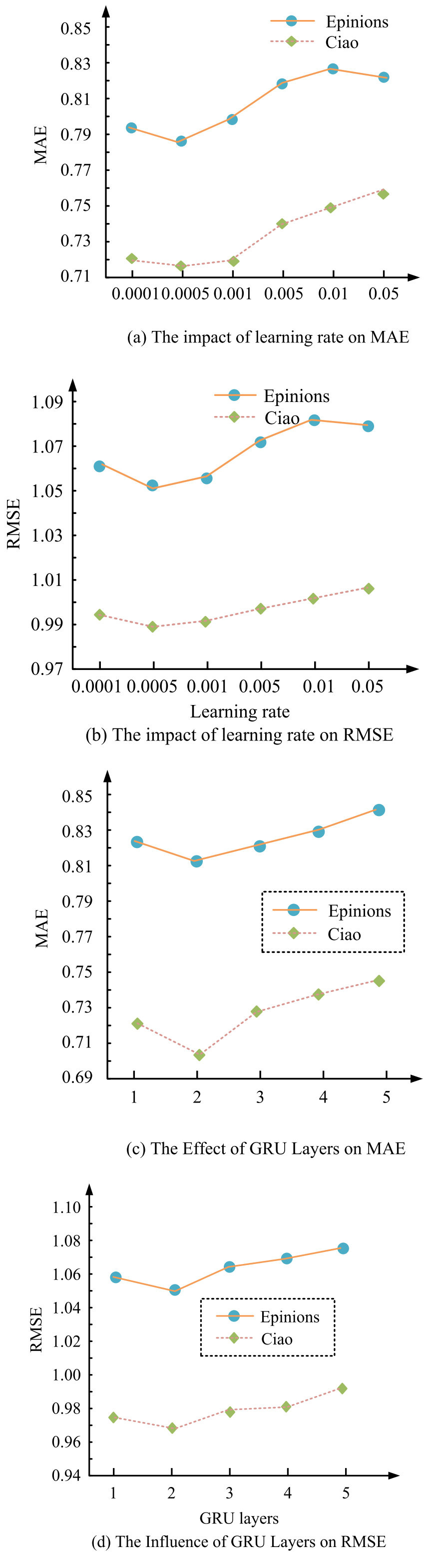


FIGURE 8. **The influence of learning rate and gru layers on MAE and RMSE.**

can quickly converge and achieve good performance in an appropriate time. For the optimization of MGRU algorithm, selecting an appropriate learning rate is crucial. It is necessary to ensure that the model can fully learn the patterns and

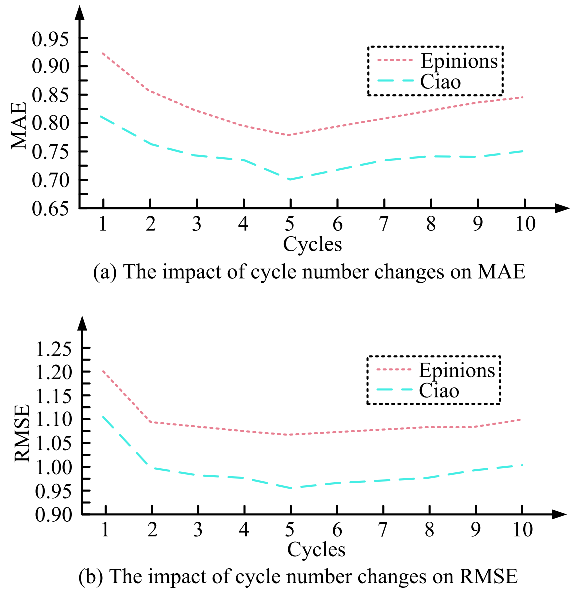
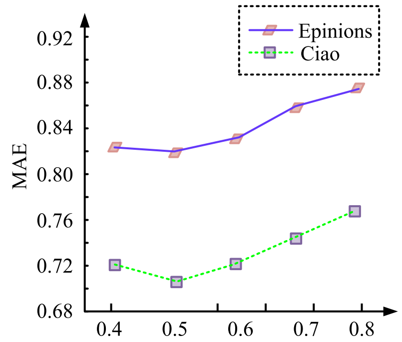


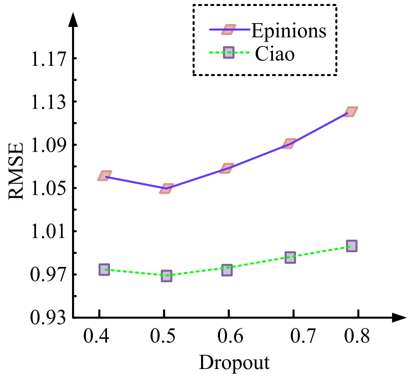
FIGURE 9. **The impact of changes in cycle numbers on MAE and RMSE.**

features of the data, while avoiding excessive correction or oscillation issues, in order to achieve the best results. These results provide useful reference for optimizing hyperparameter selection and model training of MGRU algorithm.MAE and RMSE decreased as layers increased, which indicated that the more layers, the more likely to produce overfitting phenomenon, and the fewer layers, the weaker the ability to learn time series information. When the number of layers is 2, the MAE and RMSE values of the MGRU algorithm are lower, indicating that the predicted results of the model are closer to the actual observation values and perform better. This may indicate that when the number of layers is more than 2, overfitting is prone to occur, that is, the model over learns the noise and details in the training data, resulting in weak generalization ability on new data. On the other hand, when the number of layers is less than 2, the model's temporal information learning ability may be relatively weak, making it difficult to capture complex temporal relationships in the data. Therefore, setting the number of layers to 2 in this study can achieve good performance. This setting ensures that the model has a certain ability to learn temporal information while effectively avoiding overfitting problems. This helps to improve the generalization ability and prediction accuracy of the model, providing useful reference and guidance for the analysis and prediction of time series data. In summary, setting the number of layers to 2 is to balance the temporal information learning ability and overfitting problem in the MGRU algorithm, and achieve good performance under this setting. Next, the variation of prediction error was analyzed with the number of iteration cycles, as shown in Figure 9.

Figures 9 (a) and 9 (b) respectively show the impact of the number of cycles on MAE and RMAE. From Figure 9, it can be seen that for each iteration, the vertices of the user were aggregated once. When the iteration period was 5, the optimal solution was reached. When iterations increased, overfitting



(a) The impact of Dropout on MAE



(b) The impact of Dropout on RMSE

FIGURE 10. The variation of optimal prediction error with dropout probability.

occurred. The dropout layer was set between the output end of most nonlinear activation units and the GRU layer, and the overall consistency of the probability of the dropout layer was used to regulate the dropout layer, so as to effectively suppress overfitting. The relationship between the optimal prediction error of the model and the dropout probability is shown in Figure 10.

Figures 10 (a) and 10 (b) represent the impact of dropout probability on MAE and RMAE. It can be seen from Figure 10 that when 50% probability was used as rejection number, the model can well balance prediction error and overfitting. Although the model remained unaffected with a 70% probability, its prediction errors significantly increased. Through verification, it was found that the best performance was reached when the embedding dimension was 64, the batch size of selected samples was 64, the learning rate was 0.0005, the vertical stacking layer of GRU was 2, the iteration period was 5, and the dropout probability was 50%. To further test the research algorithm, several recommendation algorithms such as PMF, SocislMF, NeuMF, DeepSoR, and GraphRec were selected and compared, and their results were analyzed. These algorithms represent different recommendation methods and technologies, with their respective advantages and applicability. Selecting several recommendation algorithms such as PMF, SocialMF, NeuMF, DeepSoR, and GraphRec for comparison and research can provide a diverse and comprehensive perspective for algorithm research in the field of recommendation systems, thereby increasing

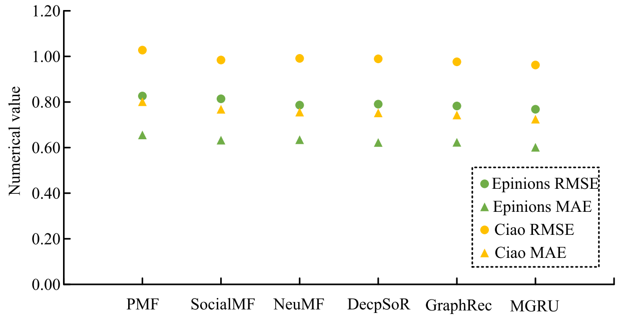


FIGURE 11. Comparison of MGRU algorithm with other models.

understanding of the performance and applicability of different algorithms, and providing useful references for the research and application of recommendation systems.In the Ciao and Epions datasets, 70% was selected for training, 20% for validation, and the remaining for the test set. The results were compared with other models, and the overall prediction error RMSE and MAE were shown in Figure 11.

In Figure 11, MGRU has a smaller error compared to other models. Among them, PMF only utilized the rating values of users and items, while SocialMF, DeepSoR, and GraphRec combined the rating values and social information, so the error of PMF was relatively small. Although NeuMF only used scoring information, it was based on neural networks and can extract more information, thus its effectiveness was also better than PMF. MGRU was used in this section, providing a new pattern that combined temporal information of user ratings with social information of users. A new method for iterating user nodes based on GRU temporal information and utilizing MHAM to analyze the impact of friends on users from various perspectives improved the overall performance.

B. EVALUATION AND PERFORMANCE ANALYSIS OF GNN ALGORITHMS INTEGRATING LSTP

The study used two common datasets to test GNNLSR's performance. The first one is the Amazon dataset, which contains purchase information for many different modules. The study randomly selected a Book module as the experimental object. In these datasets, user ratings of products were included. Because Amazon has very little data and a large number of users and projects, this study screened it as a user with at least 5 interactive projects and at least 2 interactive users during the experiment. The second dataset is Douban, which can obtain user interaction ratings and timestamps. The Douban platform is a review and recommendation platform that includes multiple fields such as books, movies, music, and more. Researchers can use this dataset to explore user ratings for different projects and the changes in ratings over time. These data are very useful for the research and evaluation of recommendation systems. The study focused on Top N recommendations, therefore two commonly used indices were used: Recall@N And Precision@N. Recall@N (Recall rate) measures how many items that have actually been rated by users in the top N recommendations of the algorithm have been successfully recalled, that

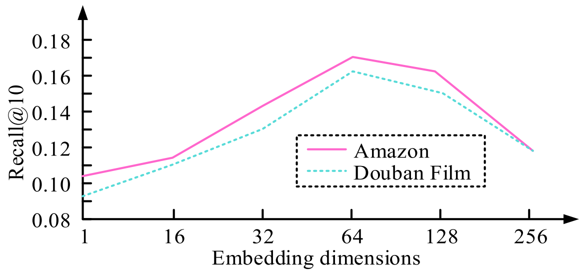


FIGURE 12. **The impact of node embedding dimension on the GNNLSR algorithm.**

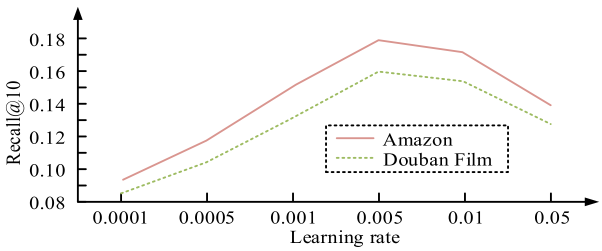


FIGURE 13. **The impact of learning rate on model results.**

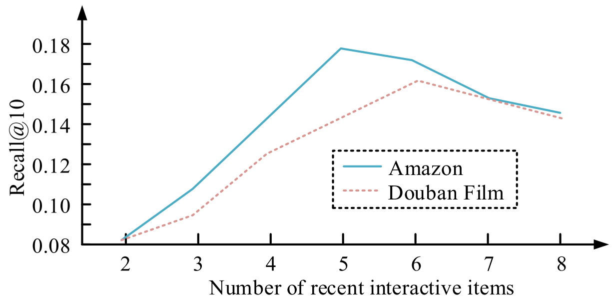


FIGURE 14. **The influence on the results of GNNLSR algorithm in the short term preference model.**

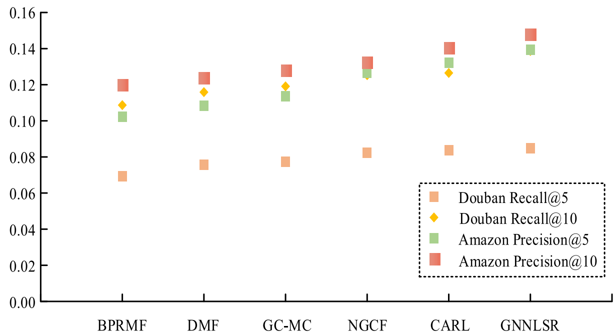


FIGURE 15. **Comparison of GNNLSR algorithm with other models.**

is, recommended. Precision@N (Accuracy) Measures how many items in the top N recommendations of the algorithm have been actually rated by the user, which is a measure of recommendation accuracy. In the experiment, $N \in \{5, 10\}$, "5", and "10" represent the recommended quantity of items. In testing, projects that did not interact with users were considered negative. On this basis, the study selected two different types of Amazon and Douban movie datasets, with 70% of the movies being the training set, 20% being the validation set, and the remaining movies being the testing set. And the impact of embedding dimension, learning rate, and the number of recent interactive items was analyzed on GNNLSR recommendation performance. After obtaining the optimal parameters, it was then compared with other recommendation algorithms. Figure 12 shows the impact of node embedding dimension on the results of the GNNLSR algorithm.

From Figure 12, it can be seen that as the embedding dimension increased, Recall@10 gradually increased, and when the embedding dimension was 64, it reached the optimal state, and then the performance decreased again. Therefore, in this experiment, the node embedding dimension was set to 64. Figure 13 shows the impact of learning rate on model results.

From Figure 13, the learning rate of 0.005 was the optimal. When the learning rate was less than 0.005, this method cannot achieve the optimal effect, and the training time of this method was too long; When the learning rate was greater than 0.005, its performance decreased. Finally, the impact of the number of items recently interacted with by users in the short-term preference model on the results of the GNNLSR algorithm was analyzed, as shown in Figure 14.

In Figure 14, a recommendation system that combined LSTP can improve recommendation accuracy. However, when the two methods experienced too many interactive items in the near future, the difference between the two models became very small, which affected recommendation effectiveness. The number of recent interactive items was set to 5. In order to further validate the effectiveness of the GNNLSR algorithm, DMF, BPRMF, and GC-MC, NGCF, GARec, which belong to graph neural network recommendation algorithms, were selected from traditional recommendation algorithms, and their performance was tested. Selecting DMF, BPRMF, GC-MC, NGCF, GARec and other recommendation algorithms for comparison and research can provide multiple comparative algorithms for the effectiveness verification of GNNLSR algorithm, further exploring the performance and applicability of GNNLSR algorithm in recommendation systems, and providing useful references for research and application in the field of recommendation systems. By comparing different types of recommendation algorithms, we can comprehensively understand their respective advantages and disadvantages, providing empirical research and theoretical basis for the development and optimization of recommendation systems. In the Amazon and Douban datasets, the study selected 70% for training, 20% for validation, and the rest as the test set, and compared them with several other models. The overall prediction comparison results are shown in Figure 15.

In Figure 15, the performance of the BPRMF based recommendation algorithm was poor. Compared with the BPRMF algorithm, the DMF algorithm used a nonlinear hierarchy to describe users' products, which can obtain more feature information and improve the efficiency of recommendation. However, recommendation systems based on graph neural networks had better performance than traditional recommendation algorithms. GC-MC was one of the earliest algorithms

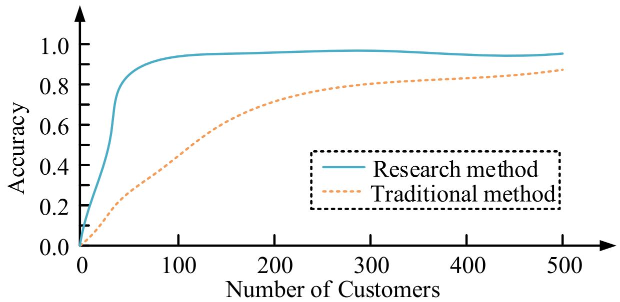


FIGURE 16. The accuracy of research methods in recommending e-commerce customers.

to apply graph autoencoders to recommendation systems. However, due to its limited ability to be predicted by a single non iterative graph neural network, it cannot effectively utilize the node information of the graph, resulting in lower efficiency compared to other graph neural networks. The NGCF method aggregated the high-order neighborhood information of nodes by introducing residual networks, and aggregated the neighborhood features of target nodes. GARec introduced an attention mechanism in recommendation, which enabled different nodes to have different attention coefficients and improved the expression ability of nodes. Therefore, its recommendation effect was superior to the other two graph neural networks. The GNNLSR algorithm proposed in the study had higher recommendation accuracy compared to other algorithms, mainly because it extracted different features based on different user preferences. This method utilized GNN to model users' short-term preferences by integrating the characteristics of project interaction and scoring. In the user long-term preference model, searching for the same user can obtain more user information. Finally, feature extraction was performed on the item to achieve a comprehensive improvement in recommendation effectiveness. Compare the research method with the traditional recommendation methods for current e-commerce customers, and compare the accuracy of the recommendation, as shown in Figure 16. Comparing traditional recommendation methods with research methods can provide an evaluation of the effectiveness and efficiency of existing recommendation methods, thereby guiding the application of recommendation systems in e-commerce customers. By comparing the recommendation accuracy of traditional and new methods, the advantages of the new method in providing personalized recommendations can be evaluated. This comparison helps to understand which method is more suitable for e-commerce customers and specific application scenarios, thus providing useful reference and empirical basis for the improvement and optimization of recommendation systems.

From Figure 16, it can be seen that the research method has a significantly higher recommendation accuracy for e-commerce customers than traditional methods, which means that this method can more accurately recommend the products they need and like to e-commerce customers, thereby promoting the development of the e-commerce industry.The graph neural network recommendation method, which integrates multi head attention mechanism and improved gated loop unit algorithm, can indeed extract behavioral data and make recommendations in the field of e-commerce, thereby promoting personalized recommendations for electronic products such as online shopping, and has a positive effect on the development of the e-commerce industry. In addition to the field of e-commerce, this method can also be applied in many other fields, including social media recommendations, news recommendations, movie and music recommendations, tourism and catering recommendations, and other neighboring areas. In social media recommendation, personalized recommendation algorithms can help users discover more interesting content and users by analyzing their behavior on social media platforms, such as likes, shares, comments, etc. In news recommendation, personalized recommendations are made in a massive amount of news content, providing relevant news reports based on users' interests and behavioral history, thereby providing a better user experience. In movie and music recommendation, personalized recommendation algorithms can provide users with movie and music recommendations that are more in line with their tastes by analyzing users' historical ratings, favorites, playback records, and other data. In the field of tourism and catering recommendations, personalized recommendations can help users discover tourist attractions, restaurants, etc. that are suitable for their taste based on their travel preferences, consumption records, and geographical location information. In summary, the graph neural network recommendation method that integrates multi head attention mechanism and improved gated loop unit algorithm has broad application potential in many fields. By utilizing user behavior data and personal preferences, these methods can provide more personalized and accurate recommendations, improve user experience, and promote the development of different industries.

V. CONCLUSION

As big data develops, various social media has brought an enormous influx of information and data, as well as information overload. Therefore, there is an urgent need for a system that can filter information based on user preferences to assist users in making better decisions. In view of this, the study first constructs a graph neural network model, and based on this, constructs GNN that integrates MHAM and GRU. To consider users' LSTP, a recommendation method for a graph neural network that combines long and short preferences is proposed. The test results indicated that it performed best when the embedding dimension, sample batch size, learning rate, GRU vertical stacking layers, iteration period, and Dropout probability were set to 64, 64, 0.0005, 2, 5, and 50%, respectively. The experimental results indicated that LSTP can achieve better recommendation results.The personalized recommendation that integrates short-term bias, attention mechanism, and other factors will be beneficial for extracting and recommending behavioral data of e-commerce customers, which in turn will be beneficial for personalized recommendation of electronic products such as online shopping and the development of the e-commerce industry. In the future, how to shorten runtime while ensuring recommendation accuracy will be further studied.

REFERENCES

[1] C. Liu, K. Wang, and A. Wu, "Management and monitoring of multi-behavior recommendation systems using graph convolutional neural networks," *Int. J. Found. Comput. Sci.*, vol. 33, pp. 583–601, Sep. 2022.

[2] X.-N. Wang and Q.-M. Tan, "DAN: A deep association neural network approach for personalization recommendation," *Frontiers Inf. Technol. Electron. Eng.*, vol. 21, no. 7, pp. 963–980, Jul. 2020.

[3] G. Babaei and S. Bamdad, "A multi-objective instance-based decision support system for investment recommendation in peer-to-peer lending," *Exp. Syst. Appl.*, vol. 150, Jul. 2020, Art. no. 113278.

[4] S. Du, Z. Chen, H. Wu, Y. Tang, and Y. Q. Li, "Image recommendation algorithm combined with deep neural network designed for social networks," *Complexity*, vol. 2021, pp. 1–9, Jul. 2021.

[5] J. Fang, B. Li, and M. Gao, "Collaborative filtering recommendation algorithm based on deep neural network fusion," *Int. J. Sens. Netw.*, vol. 34, no. 2, pp. 71–80, 2020.

[6] Z. Guo and H. Wang, "A deep graph neural network-based mechanism for social recommendations," *IEEE Trans. Ind. Informat.*, vol. 17, no. 4, pp. 2776–2783, Apr. 2021.

[7] R. Zhao, X. Xiong, X. Zu, S. Ju, Z. Li, and B. Li, "A hierarchical attention recommender system based on cross-domain social networks," *Complexity*, vol. 2020, pp. 1–13, Feb. 2021.

[8] G. Zhu, Y. Wang, J. Cao, Z. Bu, S. Yang, W. Liang, and J. Liu, "Neural attentive travel package recommendation via exploiting long-term and short-term behaviors," *Knowl.-Based Syst.*, vol. 211, Jan. 2021, Art. no. 106511.

[9] L. Huang, M. Fu, F. Li, H. Qu, Y. Liu, and W. Chen, "A deep reinforcement learning based long-term recommender system," *Knowl.-Based Syst.*, vol. 213, Feb. 2021, Art. no. 106706.

[10] H. Zhu, "Research on human resource recommendation algorithm based on machine learning," *Sci. Program.*, vol. 2021, pp. 1–10, Aug. 2021.

[11] Q. Yuan, "Network education recommendation and teaching resource sharing based on improved neural network," *J. Intell. Fuzzy Syst.*, vol. 39, no. 4, pp. 5511–5520, Oct. 2020.

[12] Z. Liu, T. Zhao, H. Han, and C. Zhang, "Personalized tag recommendation based on convolution feature and weighted random walk," *Int. J. Comput. Intell. Syst.*, vol. 13, no. 1, pp. 24–35, Jan. 2020.

[13] M. Chen, J. Cheng, G. Ma, L. Tian, X. Li, and Q. Shi, "Servicecomposition recommendation method based on recurrent neural network and Naive Bayes," *Sci. Program.*, vol. 2021, pp. 1–9, Oct. 2021.

[14] M. Bunc, S. Hadži, C. Graf, M. Boncina, and J. Lah, "Aggregation time machine: A platform for the prediction and optimization of long-term antibody stability using short-term kinetic analysis," *J. Medicinal Chem.*, vol. 65, no. 3, pp. 2623–2632, Feb. 2022.

[15] G. Xue, C. Qi, H. Li, X. Kong, and J. Song, "Heating load prediction based on attention long short term memory: A case study of Xingtai," *Energy*, vol. 203, Jul. 2020, Art. no. 117846.

[16] A. Salamat, X. Luo, and A. Jafari, "HeteroGraphRec: A heterogeneous graph-based neural networks for social recommendations," *Knowl.-Based Syst.*, vol. 217, Apr. 2021, Art. no. 106817.

[17] X. Wu, Q. Li, X. Li, V. C. M. Leung, and P. C. Ching, "Joint long-term cache updating and short-term content delivery in cloud-based small cell networks," *IEEE Trans. Commun.*, vol. 68, no. 5, pp. 3173–3186, May 2020.

[18] X. Yuan, L. Li, Y. Wang, C. Yang, and W. Gui, "Deep learning for quality prediction of nonlinear dynamic processes with variable attention-based long short-term memory network," *Can. J. Chem. Eng.*, vol. 98, no. 6, pp. 1377–1389, Jun. 2020.

[19] J. Zhang, X. Liu, X. Zhou, and X. Chu, "Leveraging graph neural networks for point-of-interest recommendations," *Neurocomputing*, vol. 462, pp. 1–13, Oct. 2021.

[20] N. Shakeel, P. Teradata, and S. Shakeel, "Context-free word importance scores for attacking neural networks," *J. Comput. Cognit. Eng.*, vol. 1, pp. 187–192, Sep. 2022.

FANG LIU was born in March 1978. She received the bachelor's degree in computer information management from Hebei University, in July 2001, and the master's degree in computer application technology from Shanxi University, in July 2010. She is currently with the Shijiazhuang Institute of Technology, majoring in software technology development and artificial intelligence direction. She is also the Executive Dean of the Internet Application Technology College. She is also an associate professor. She has published more than 20 academic articles, including two core articles in Chinese, and participated in four scientific research projects. She is the chief of three textbooks.

JUAN WANG was born in February 1982. She received the bachelor's degree in computer science and technology from the PLA Shijiazhuang Army Command College, in July 2003, and the master's degree in computer application systems from the Hebei University of Science and Technology, in July 2007. She is currently with the Shijiazhuang Institute of Technology, majoring in computer software technology. She is a lecturer. She has published seven academic articles and participated in one scientific research project.

JUNYE YANG was born in August 1982. She received the bachelor's degree in computer and application from the Hebei University of Science and Technology, in 2005, and the master's degree in computer software and theory from the Shenyang University of Chemical Technology, in 2010. She has completed the Junior College Education. She is currently with the Shijiazhuang Institute of Technology, majoring in computer technology. She is also an associate professor. She has published more than 20 academic articles, including two Chinese core articles, and participated in seven scientific research projects.