Research on Enhanced Multi-head Self-Attention Social Recommendation Algorithm Based on Graph Neural Network

Yue Teng, Kai Yang

Abstract—In today's society, communication among people has become more frequent and extensive due to the rapid development of science, technology, and the Internet. This vast communication occurs in real life and virtual online worlds. However, with the overwhelming amount of information available, people often struggle with finding and accessing the information they need, leading to frustration. To address this challenge, we conducted a study to extract valuable information and insights from users' social relationships. While existing models can help us understand and analyze users' social relationships to some extent, there are still significant challenges in extracting valuable insights. Therefore, we conducted in-depth research to improve the GraphRec model and created a new social recommendation model called the CWYGNN model. The CWYGNN model is unique because it can identify deeper connections in users' social interactions through in-depth learning of the social sequences between users. This is important for understanding users' behavioral patterns, predicting their needs, and providing personalized services. To validate the effectiveness of the CWYGNN model, we conducted experiments and comparisons on several public datasets. The experimental results show that the CWYGNN model outperforms similar approaches in processing user social relationships and extracting valuable insights. This result provides a new way of thinking and methodology to help us better understand and serve users and ultimately enhance their overall experience.

Index Terms—**Enhanced multi-head self-attention, Graph neural network, Recommendation system, Social recommendation**

I. INTRODUCTION

In recent times, recommendation systems that utilize social networks have gained significant popularity. These systems assume that people's preferences can be influenced by those in their social circle, including their closest acquaintances, which is supported by social relevance theories. SoRec [1] proposes joint factor decomposition, which uses ratings and social relationships to break down potential matrices of standard user features. TrustMF is a recommendation algorithm that combines user ratings and social network information through a matrix. Factorization, resulting in more accurate recommendations. This model considers the trust and trusted users' relationship with users and items, ultimately improving recommendations' precision. SoDimRec is a recommendation model that enhances the accuracy of recommendations by utilizing social dimensions. The SoDimRec [2] model uses a graph to represent user relationships. A node represents each user, and edges represent the connections between them. The model employs matrix factorization techniques to learn the representations of nodes and edges. This way, it can predict user ratings for items. A Graph Neural Network (GNN) model was selected and optimized to improve the current social recommendation algorithms. The existing attention mechanism was enhanced by integrating linear transformations and feed-forward neural networks. As a result, the multi-head attention mechanism is improved [3]. Afterward, the newly established Graph Neural Network and improved Multi-head Self-attention Network were used to gather the latest information between the user and the object. The main aim was to gain a deeper understanding of user interests and track real-time changes in user behavior.

At the prediction layer, the system calculates optimal mean square error and mean absolute error values for each item based on the learned information to provide suitable recommendations.

The CWYGNN model consists of three main components, as shown in Figure 1. Firstly, the user's social data is converted into a graph structure data. This data is then input into the Graph Neural Network to perform complex transition pattern calculations, which helps determine user interest changes. Secondly, the multi-head attention mechanism is employed to learn the user's social sequence further and uncover hidden information within their social relationships.

This paper proposes a novel solution to the challenges faced by social recommendation systems. The solution is a social recommendation model based on a Graph Neural Network (GNN) with enhanced self-multi-head attention (CWYGNN). The CWYGNN model enhances and optimizes existing social recommendation models. It addresses the issue where current models can only capture users' short-term interests and fail to comprehend their long-term preferences. Firstly, the paper introduces a new GNN model called CWYGNN, which can effectively delve into user communication information, capturing dynamic user interests. This is achieved by integrating other GNN models and improving self-multi-head attention. Secondly, new training models have been incorporated into the original model to capture dynamic user information better,

Manuscript received October 25, 2023; revised May 16, 2024.

Yue Teng is a postgraduate student at the School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, 114051, China (e-mail:1127743361@qq.com).

Kai Yang is a Professor at the School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, 114051, China (corresponding author, phone:86-15124122757; e-mail: asyangkai@126.com).

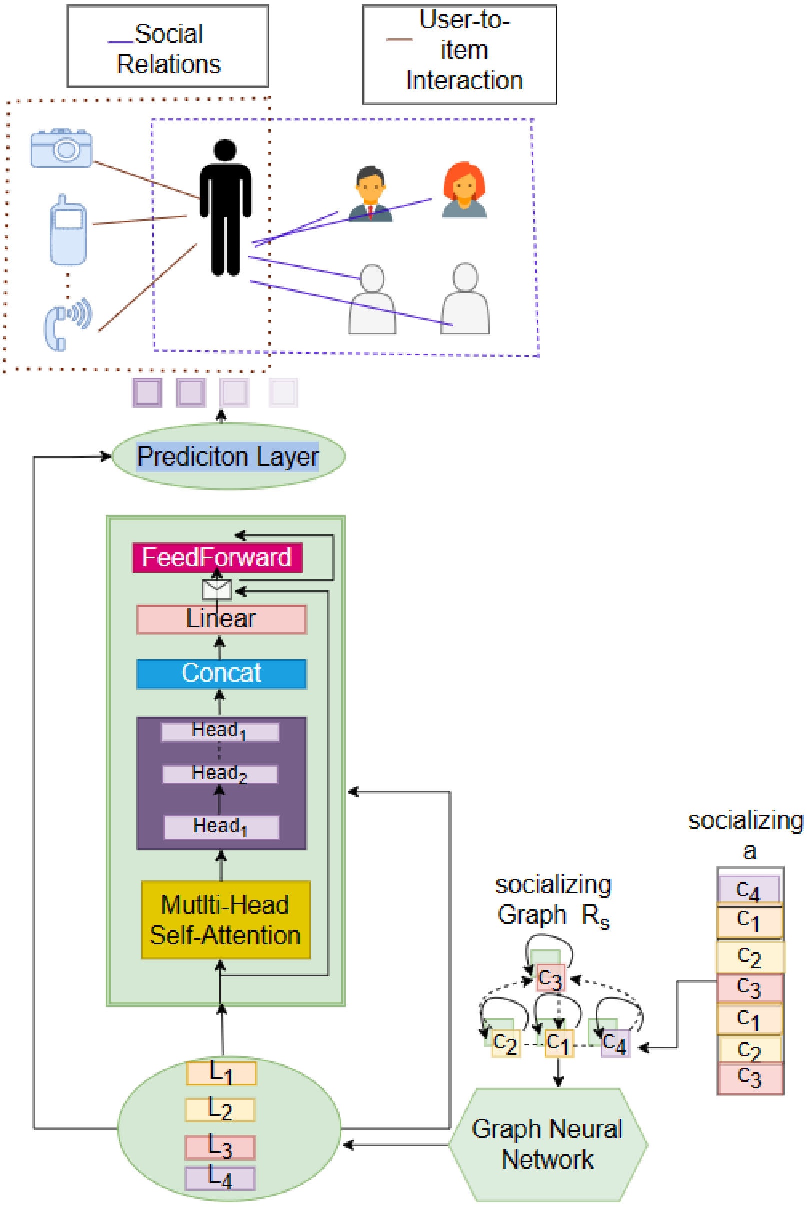

Fig. 1 Social recommendation - graph data. It contains eight graphs. Firstly, from right to left, the first two are the user social data graph and the neural network graph. The following two are the transition graph of the graph neural network and the attention self-attention learning graph. Then, there is the prediction layer graph, user-item graph, and user-user social graph.

accounting for real-time interest changes during user interactions. Thirdly, an enhanced multi-head self-attention mechanism has been integrated to thoroughly explore the latent information within communication items and acquire more accurate local details. This allows the model to understand better and capture shifts in user interests. To validate the feasibility and effectiveness of the proposed CWYGNN model, the researchers conducted comprehensive experiments using two real-user public datasets from Ciao and Epinions. The experimental results indicate that the CWYGNN model outperforms other similar models significantly. This result confirms the effectiveness and practicality of the proposed model.

The paper's innovation can be mainly identified through the following aspects:

1) A new Graph Neural Network model named CWYGNN is proposed to capture dynamic user information more deeply.

2) The GraphSage network model is integrated into the original model, combining the algorithm of the GraphSage model with that of the original model to enhance the capturing of user dynamics.

3) Communication items are thoroughly analyzed for hidden information through an enhanced multi-head self-attention mechanism to acquire precise local details.

4) Extensive experiments have been conducted to validate the feasibility and effectiveness of the CWYGNN model proposed.

In our report, we will follow a sequence of events. Firstly, we will examine the framework proposed in Section 2. Secondly, we will conduct extensive experiments on two real-world datasets to validate our method's practical effectiveness and feasibility in Section 3. Section 4 will outline other frameworks similar to the one we propose. Finally, we will summarize our findings and discuss potential future research directions and goals in Section 5, especially in social recommendation systems. We must examine how to provide clearer and more understandable explanations and illustrations to enhance users' trust and acceptance of recommender systems and provide them with a better recommendation experience.

II. Proposed Framework

This section introduces definitions and symbols, elaborates on the modeling process and its components, and discusses methods for learning model parameters.

TABLE I
NOTATION

Symbol	DEFINITIONS AND DESCRIPTIONS
C	The collection of all social projects.
C_i	The projects that the user has clicked on in social S.
L	Control weights.
L_i	The initial embedding vector of c_i .
l	The index of the hidden layer.
D_i	The two columns in the adjacency matrix A that correspond to node c_i .
d_i^t	The result of the bidirectional transmission and interaction of node information.
w_i^t	Represents the update gate.
e_i^t	Represents the reset gate.
$\gamma(\bullet)$	Represents the Sigmoid function.
$\otimes$	The dot product operator.
n	Represents the number of attention heads.
F^o	Represents the projection matrix.
$F_z F_r F_0$	Represents a parameter matrix that can be learned.
$M_z M_r M_o$	Matrix that can be learned.
$F_1 F_2$	Weight matrix.
$e_1 e_2$	Bias vector.
$F^Q F^K F^V$	The projective matrix that can be learned.
b_{cw}	The rating value of user j_c for item i_w .
$b_{cw}^{'}$	The predicted rating value of user j_c for item i_w .
g_i	Represents the predicted score of candidate item c_i .

A. Construct a social graph.

Let $C=\{c_1,c_2,\cdots,c_m\}$ represent all unique items associated with a social user, referred to as an itemset. Let m represent the total number of items. The unknown user's social sequence is defined by $a=\{c_1,c_2,\cdots,c_n\}$, where $c_i \in C$ refers to the a item that the user clicked on during their social interaction. Let's take the social sequence $a=\{c_1,c_2,\cdots,c_n\}$ as an example. Construct the social sequence a as a weighted directed social graph $R_s=(C_s,B_s)$, where $R_s \in R$ and R represent the set of all social graphs. In the social graph R_s, the node set C_s encompasses all the item nodes within this social network. Each node is representative of an item $C_i \in C_s$. The edge set B_s represents the set of all outgoing edges. Each directed edge $(c_i,c_{i+1}) \in B_s$ signifies that the user clicked on an item c_i and subsequently clicked on the item c_{i+1} in social a.

After completing the processing above steps, the social graph is exported to the model to extract hidden information. We aim to predict the next item c_{n+1} that users will click on based on the social sequence a. In this scenario, the user's social sequence is used as an input to the model. The project achieves optimal mean square and average absolute error through multiple iterations of the learning and prediction layers.

B. The embedding vector of the social graph

This section focuses on integrating the social graph into the graph neural network to extract embedded information about global projects. The social graph is input into a graph neural network to derive an embedding representation for the international item.

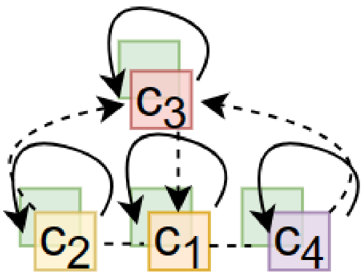

Fig. 2 Social graph and CWYGNN network structure.

Long-term user interests can be captured by analyzing patterns of item transitions that encompass the temporal sequence inherent in the user's interactions. To learn the embedding vectors of the nodes in the graph, a gated graph neural network is chosen, a classic spatial message-passing model that utilizes GRU to facilitate communication between nodes and achieve iterative information updates. The update function for the node vector in the conversation graph can be represented as follows:

$$d_i^t = D_{I:}[l_1^{t-1},\ldots,l_n^{t-1}]^T \mathrm{L}+\mathrm{b} \tag{1}$$

$$w_i^t = \gamma(F_z d_i^t + M_z l_i^{t-1}) \tag{2}$$

$$e_i^t = \gamma(F_r d_i^t + M_r l_i^{t-1}) \tag{3}$$

$$\mathbf{l}_l^t = \left(\mathrm{F_O d_i^t} + \mathrm{M_o}(\mathrm{e_i^t} \otimes l_i^{t-1})\right) \tag{4}$$

$$l_i^t = (1-w_i^t) \otimes l_i^{t-1} + w_i^t \otimes \mathbf{l}_l^t \tag{5}$$

Take the social sequence $a=[c_{4,}c_1,c_2,c_3,c_1,c_2,c_3]$ as an illustration. After multiple iterations, the final vector of each node is obtained using Equations 2 to 5, which is analogous to the calculation process of the GRU and updated through generating and forgetting control information.

C. Enhance the bull self-attention network

We have developed a method to improve multi-head self-attention networks[3] by adding multiple attention heads to different subspaces. This allows us to extract and combine information from each subspace to understand user interests better. By considering both the long-term and short-term interests of the user, our joint prediction presents a list of items that may capture their interest.

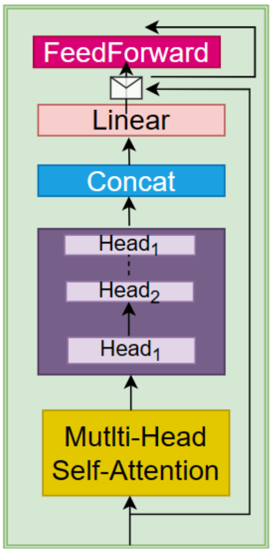

Fig. 3 CWYGNN network architecture.

The self-attention mechanism is a process that involves mapping queries and key-value pairs to outputs. This is followed by calculating the weighted sum of values using previously obtained output. The corresponding key and query determine the weight of each value. The weights are calculated based on how similar they are to each other. The embedding matrix of each user's social interactions can be obtained through multi-layer computation of graph neural networks E . The self-attention network receives the embedding matrix and produces matrices Q, K, and V through multiple linear transformations. The calculation formula for this process is expressed as follows [3]:

$$Q=Linear(E)=EF^{Q} \tag{6}$$

$$V=Linear(E)=EF^{V} \tag{7}$$

$$K=Linear(E)=EF^{K} \tag{8}$$

$$Attention(Q,K,V)=softmax\left(\frac{QK^{T}}{\sqrt{d_k}}\right)V \tag{9}$$

$F^{Q},F^{K},F^{V}\in R^{d\times d}$ represents a learnable matrix, and the embedding vectors of all nodes in the social network are composed of $E=\{e_1^t,e_2^t,\cdots,e_n^t\}$ to represent the embedding matrix of the social network [4]. The inner product of the vector can be calculated by multiplying it by "and." Next, the softmax function is used to calculate the importance coefficient of each item relative to the others. However, it should be divided by the square root to prevent excessive inner products d_k . In some projects, dependencies are learned by assigning weights through self-attention in an adaptive manner.

The multi-head self-attention [5] network is a crucial component in natural language processing. Its primary function is to identify long-distance relationships within input sequences. The network works by dividing the input sequence into multiple segments and processing each simultaneously. Each segment is processed by a unique "head" that transforms it into a distinct representation vector. These vectors are then combined to produce the final output. One significant advantage of the multi-head self-attention network is its ability to learn complex patterns within the input sequence. By considering multiple pieces of information at once, the network can comprehend and represent data more profoundly. Additionally, the number of "heads" can be adjusted to increase or decrease the model's complexity and computational efficiency. This flexibility allows the multi-head self-attention network to excel in various natural language processing tasks. The calculation formula for the network is as follows:

$$A=MultiHead(Q,K,V)=Concat(head_1,\cdots head_n)F_o \tag{10}$$

$$head_i=Attention(NF^{Q},NF^{K},NF^{V}) \tag{11}$$

Where $F^{O}\in R^{d\times d}$ represents the projection matrix and n represents the number of attention heads.

The initial enhanced multi-head self-attention module [6] combines the embedding information of all items in the original data. A state-of-the-art multi-head self-attention network was created by stacking multiple enhanced multi-head self-attention modules to gain a deeper understanding of session data. This network takes the output of the previous module as input for the next one, allowing for multi-level learning. The b module can be defined as follows:

$$F^{b}=CWYN(F^{b-1}),\forall i\in 1,2,\cdots,n. \tag{12}$$

After conducting multiple experiments, it was determined that the model with b=2 performed better than those with b=1, 3, 4, and so forth.

D. Point feed-forward networks

A pointwise feedforward network [7] is added after the multi-head self-attention network. It includes two linear transformations and a ReLU activation function to increase the model's nonlinear capabilities. The specific definitions are as follows:

$$\mathrm{FFN(A)=ReLU(AF_1+b_1)F_2+b_2} \tag{13}$$

To prevent overfitting, optimize the model using the Dropout technique [7] and mitigate gradient vanishing with residual connections, which improves robustness. The calculation method is as follows:

$$\mathrm{A'=A+Dropout(MultiHead(A))} \tag{14}$$

$$\mathrm{F=A'+Dropout(FFN(A'))} \tag{15}$$

$$F=CWYN(H) \tag{16}$$

where F_1, F_2 represent weights and e_1, e_2 represent bias vectors. The augmented multi-head self-attention network is defined as a whole for convenience and brevity.

E. Graphsage architecture

The Graphsage architecture [8-9], designed to handle large datasets, generates embedding vectors that can be applied to downstream tasks. A new class called NeighborAggregator has been added to the existing Graphsage architecture. Its purpose is to integrate the characteristics of the neighbor node and convert them into an embedded representation of the target node. This is done by calculating the aggregate value of the neighbor node features based on the user-specified aggregation method (mean, sum, or max) and then transforming these features into a hidden layer representation through a linear transformation. The Graphsage architecture is known for its superior training speed compared to conventional neural network models. The calculation method can be explained as follows:

Average [10]:

$$Agg_{v_i}^{mean} = \sigma(MEAN(Fv_j + e)), \forall v_j \in Neighbor^{\leq 1}(v_i) \quad (17)$$

Pooling [10]:

$$Agg_{v_i}^{pool} = MAX\{\sigma(Fv_j + e)\}, \forall v_j \in Neighbor^{\leq 1}(v_i) \quad (18)$$

Where F and e are the parameters to be learned, and $Neighbor(v_i)$ represents all neighbors within a hop of the node v_i.

F. Evaluation forecasts

In this section, we propose a task for learning model parameters, specifically for rating prediction recommendation tasks involving item ranking and rating prediction. In this study, we plan to use the CWYGNN model. The potential factors $[z_c \Leftrightarrow g_w]$ of the user and the project will be integrated and inputted into a multi-layer perceptron (MLP) for prediction to achieve our objective. The calculation method can be explained as follows:

$$y = [z_c \Leftrightarrow g_w] \quad (19)$$

$$y_2 = \sigma(F_2 \cdot y_1 + e_2) \quad (20)$$

$$\cdots$$

$$y_{l-1} = \sigma(F_l \cdot y_{l-1} + e_l) \quad (21)$$

$$b_{cw}^{'} = F^T \cdot y_{l-1} \quad (22)$$

Here l is the index of the hidden layer, and b_{cw} is the predicted rating from j_c to i_w.

z_c represents the user latent factor of the user's j_c, which is combined from the project space z_c^C and social space z_c^s.

z_c^C the term space user latent factor from the item set $B(\mathrm{c})$ of the user j_c.

z_c^s social space user latent factor from user j_c social friend a_i.

The calculation methods are as follows:

$$W_{ei} = y_\upsilon([h_a \odot p_r]) \quad (23)$$

The perception representation of modeling the exchange of opinions is represented as W_{ei}, which combines h_a and p_r embedding opinions. y_υ can be combined with interactive information and opinion information.

$$z_c^C = \sigma(F \cdot \left\{ \sum_{i \in B(c)} i_c w_{ei} \right\} + e) \quad (24)$$

Take the element-by-element average of the vectors $\{w_{ei}, \forall_i \in B(c)\}$. This mean-based aggregator can be conceptualized as a linear approximation of local spectral convolution.

Where i_c is fixed to $\frac{1}{|B(c)|}$, which is utilized for items in all mean-based aggregators. This approach assumes that all interactions impact the user's comprehension equally. However, this may not always be accurate. It is advisable to assign weights to each interaction so that they can contribute differently to the user's underlying factors.

$$z_c^S = \sigma(F \cdot \left\{ \sum_{o \in A(i)} \partial_i z_o^C \right\} + e) \quad (25)$$

For items in all mean-based aggregators, the element-by-element mean of $\{z_o^C, \forall_i \in A(i)\}$ vectors is taken, with $A(i)$ fixed at $\frac{1}{|A(i)|}$ .where z_o^C represents social attention, F and e represent attention weights, $b_{cw}^{'}$ represents predicted ratings from j_c to i_w, and l represents the index of the hidden layer.

G. Model training

An appropriate objective function needs to be chosen to optimize the parameters of the CWYGNN model. As the primary goal is rating forecasting, the commonly employed objective function can be formulated as follows:

$$Loss = \frac{1}{2|O|} \sum_{c,w \in O} (b_{cw}^{'} - b_{cw})^2 \quad (26)$$

$|O|$ represents the number of ratings observed, while b_{cw} signifies the actual rating of the item w by the user c. The optimization method chosen for the objective function is the Adam optimizer instead of the typical stochastic gradient descent (SGD) method. The Adam optimizer offers several advantages, making it an ideal choice for optimizing algorithms. It comes with an adaptive learning rate feature that adjusts the learning rate automatically based on various parameters. Each parameter can be assigned a unique learning rate better suited to its specific characteristics and how often it changes. Unlike SGD, which uses fixed learning rates, the Adam optimizer can react more flexibly to parameter changes. Taking small steps helps prevent overfitting problems and maintains the model's generalization ability.

III. EXPERIMENT

A. Experiment setup

Data set. For this particular experiment, we have selected two publicly available datasets, namely Ciao and Epinions, to conduct simulation experiments to evaluate the model's performance on the dataset. Ciao is an open-source dataset that is widely used and provides a vast amount of user social information from popular social networking sites (you can find more information at http://www.ciao.co.uk). On the other hand, Epinions is a dataset that comprises millions of users who can engage in simultaneous online commenting. It contains the social information of numerous users and originates from public websites (you can find more information at www.epinions.com). As a result, these data offer a wealth of rating and social information, with the evaluation scales ranging from 1 to 5. These five numbers were randomly arranged to initialize the opinion embedding. Specifically, five distinct embedding vectors were selected. To access statistical data for these two datasets, kindly refer to TABLE II.

TABLE II
DSTASET STATISTICS

Data set	Ciao	Epinions
Number of users	7,317	18,088
Number of projects	10,4975	261，649
Number of ratings	283,319	764,352
Density (nominal)	0.0368%	0.0161%
Social contact	111,781	355,813
Density(social relations)	0.2087%	0.1087%

Evaluation indicators. To ensure that the recommendation algorithm is of high quality, we evaluate the accuracy of its predictions using two commonly used metrics: Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). The prediction accuracy is higher when the MAE and RMSE values are more minor. Even small improvements in RMSE or MAE can significantly enhance the quality and effectiveness of previous recommendations.

Baseline. Three different recommendation systems were compared to ensure the accuracy and reliability of the evaluation results for CWYGNN. These systems include the traditional recommendation system, the traditional social recommendation system, and the recommendation system based on a deep neural network. Representative classical baseline models were selected for each of these methods, and their performance is discussed below.

•PMF [11]: Probabilistic matrix factorization uses a Gaussian distribution to determine the characteristics of users and items based on their rating matrix.

•SoRec [12]: Social recommendation algorithms extract useful features by combining user ratings and social relationships through collaborative factorization.

•SoReg [13]: Social network information is converted into regularization terms through social regularization to restrict matrix factorization algorithms.

•SocialMF [14]: This recommendation system incorporates trust information and propagates it using matrix factorization.

•TrustMF [15]: This method utilizes matrix factorization to project users into two lower-dimensional spaces. The trust network is divided into a trust space and a trustee space based on the direction of the trust.

•NeuMF [16]: This method uses a neural network-based matrix factorization model initially designed for recommendation ranking but later modified with a squared loss function based on predicted ratings.

•DeepSoR [17]: This model uses deep neural networks to extract user features from social relationships and integrates them into a probabilistic matrix factorization model for rating predictions.

•GraphRec [18]: This model combines a state-of-the-art recommendation system with a graph neural network architecture. It uses network embedding technology to reduce the dimensionality of the original input feature vector to a concise and dense vector. This simplifies the model's training process.

PMF and NeuMF are collaborative filtering-based models that generate predictions without considering social network information. In contrast, the other models belong to the domain of social recommendations. CWYGNN was also compared to two advanced neural network-based social recommender systems, DeepSoR and GraphRec.

We used a percentage of the dataset for training and validation purposes, with x% of the data being used for learning parameters (Xi) and (1-x%)/2 being utilized for hyperparameter tuning. The remaining (1-x%)/2 data was used to evaluate the final performance. The value of x varied from 80% to 60%. For the embedding dimension (d), we experimented with the values 8, 16, 32, 64, 128, and 256. The batch size and learning rate were also tuned in the 32, 64, 128, and 512 ranges, and 0.0005, 0.001, 0.005, 0.01, 0.05, and 0.1, respectively. To mitigate overfitting, we used the Adam optimizer to optimize these parameters. We considered the experimental data from previous comparisons of the GraphRec model and the following comparative experiments of the CWYGNN model.

B. Recommend a performance comparison of systems

First, we have compared the performance of all the proposed methods. In Table 3, you will find a detailed overview of the overall rating prediction errors (including RMSE and MAE) for the recommended techniques on the Ciao and Epinions datasets. Below are the main observations:

Four recommendation models, namely SoRec, SoReg, SocialMF, and TrustMF, have performed better than PMFs. These models use a technique called matrix factorization, and they incorporate both rating data and social network information. In contrast, PMFs only rely on rating data. Based on this comparison, it can be concluded that using social network information can improve the accuracy of recommendations.

•NeuMF is superior to PMFs in recommendation systems because it solely relies on rating information. It is worth noting that NeuMF uses a neural network architecture, highlighting the strength of this approach for recommender systems.

•DeepSoR and GraphRec outperform SoRec, SoReg, SocialMF, and TrustMF by integrating rating data and social network information. Additionally, both DeepSoR and GraphRec are based on neural network architecture, highlighting the significant potential of neural network information. Additionally, both DeepSoR and GraphRec are based on neural network architecture, highlighting the significant potential of neural network models in recommender systems.

TABLE III
PERFORMANCE COMPARISON OF DIFFERENT RECOMMENDER SYSTEMS

Training Ciao (60%)	MAE	RMSE	Training Ciao (80%)	MAE	RMSE
PMF	0.9520	1.1967	PMF	0.9021	1.1238
SoRec	0.8489	1.0738	SoRec	0.8410	1.0652
SoReg	0.8987	1.0947	SoReg	0.8611	1.0848
SocialMF	0.8353	1.0592	SocialMF	0.8270	1.0501
TrustMF	0.7681	1.0543	TrustMF	0.7690	1.0479
NeuMF	0.8251	1.0824	NeuMF	0.8062	1.0617
DeepSoR	0.7813	1.0437	DeepSoR	0.7739	1.0316
GraphRec	0.7540	1.0093	GraphRec	0.7387	0.9794
CWYGNN	**0.7520**	**1.0056**	**CWYGNN**	**0.7267**	**0.9631**

Training Epinions (60%)	MAE	RMSE	Training Epinions (80%)	MAE	RMSE
PMF	1.0211	1.2739	PMF	0.9952	1.2128
SoRec	0.9086	1.1563	SoRec	0.8961	1.1437
SoReg	0.9412	1.1936	SoReg	0.9119	1.1703
SocialMF	0.8965	1.1410	SocialMF	0.8837	1.1328
TrustMF	0.8550	1.1505	TrustMF	0.8410	1.1395
NeuMF	0.9097	1.1645	NeuMF	0.9072	1.1476
DeepSoR	0.8520	1.1135	DeepSoR	0.8383	1.0972
GraphRec	08441	1.0878	GraphRec	0.8168	1.0631
CWYGNN	**0.8328**	**1.0705**	**CWYGNN**	**0.8118**	**1.0565**

•During initial testing, GraphRec displayed satisfactory performance, indicating that GNN has excellent potential in learning graph data representation by effectively combining node information and topology.

•The CWYGNN proposed method has consistently outperformed all the other techniques used as baselines. Unlike DeepSoR and GraphRec, our model incorporates more advanced components that enable it to integrate rating data and social network information effectively. In the following chapters, we detail the contribution of each model component and propose a suitable framework.

To summarize, the comparative results indicate that:

1) Using social network information can be beneficial in improving the precision of recommendations.

2) Incorporating neural network models can significantly enhance the efficiency of recommendation systems.

3) The proposed framework outperforms representative baselines.

C. Model analysis

This section will explore how model components and hyperparameters impact outcomes.

The text explores the impact of social networks and user opinions. It proposes an efficient framework and model components to incorporate all social network data and user opinions into project information. The text also summarizes how CWYGNN works and compares it with two variations, CWYGNN-SN and CWYGNN-Opinion. In addition, it provides definitions of both variants:

•CWYGNN-SN: The social networking information of users is not included in CWYGNN. This variant solely employs the item space latent factor z_c^C to represent the user's potential factor, disregarding the user latent factor z_c^s in the social space.

•CWYGNN-Opinion: This paragraph discusses a variant in the learning process that doesn't consider the user's opinion on an item. Instead, it focuses only on the user and project latent factors in the project space without considering the user's perspective on how they interact with the project. Figure 4 shows how CWYGNN and its two variants performed on the Ciao and Epinions datasets. After analyzing the results, the following conclusions were drawn.

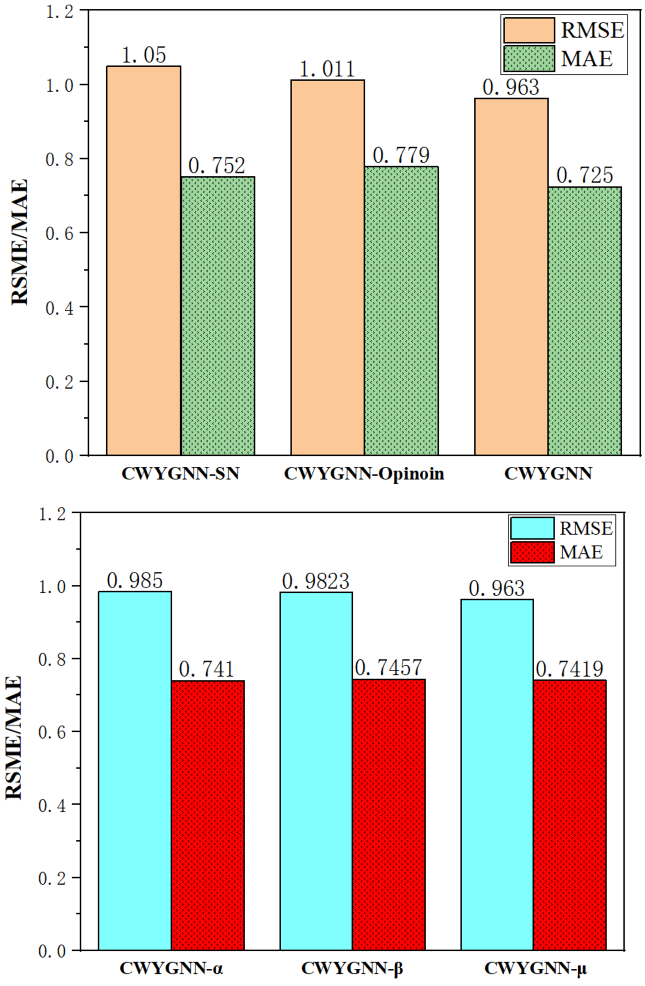

Fig.4 The impact of social network and user information on the ciao and epinions datasets.

•Social Network Information: To investigate the impact of social network information on recommender systems, the study found that CWYGNN is more effective than CWYGNNSN. These results confirm the significance of social network information in enhancing and refining recommendations.

•Opinions are crucial during interactions and play a vital role in rating predictions. When opinion information is missing, the accuracy of rating predictions decreases substantially. For example, on the Ciao and Epinions datasets, the average relative error was 3.50% and 2.64%, as measured by RMSE, and 5.84% and 5.02%, as measured by MAE, respectively. This highlights the significance of user reviews in providing valuable insights into the underlying factors of users or projects, thereby enhancing the effectiveness of recommendation systems.

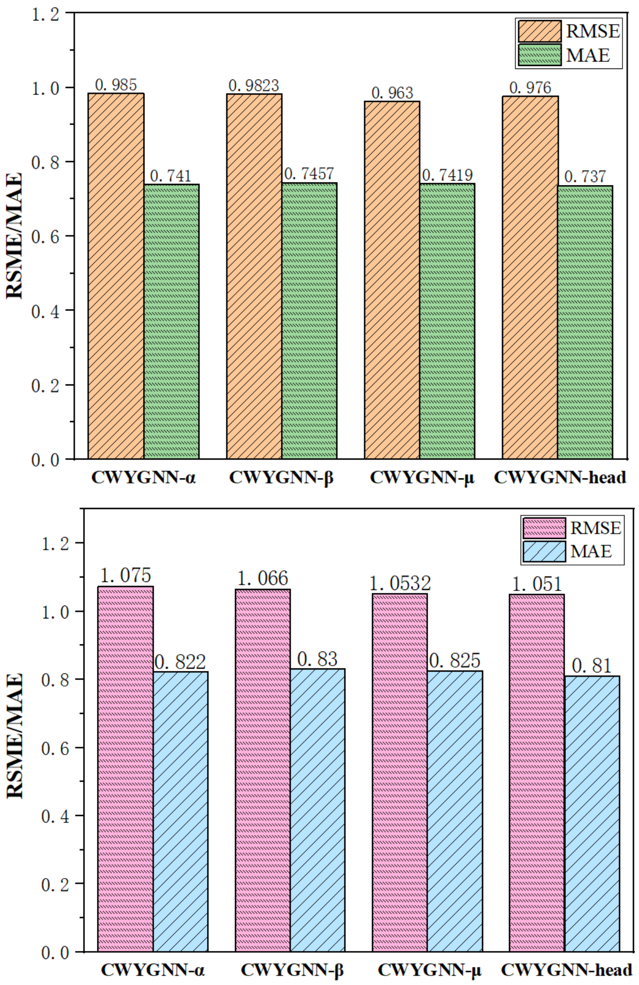

Fig.5 Impact of attention mechanisms on the ciao and epinions datasets.

We discuss the impact of using a multi-head attention mechanism to enhance the performance of our model. To improve the model further, we integrate the multi-head self-attention mechanism and conduct experiments to compare it with the previous version that did not use this mechanism. The results of this comparison are illustrated in Figure 5.

The study compared CWYGNN and its four variants: CWYGNN-α, CWYGNN-β, CWYGNN-head, and CWYGNN-μ. The four variants are defined as follows:

CWYGNN-α: This is called project attention, often overlooked when aggregating the idea-aware interactive representation of items. The mean aggregation function aggregates items and represents user-latent factors in the project space.

CWYGNN-β: This is expressed as social attention. CWYGNN's social attention α is disregarded in this variant when aggregating user neighbors. This variant utilizes a mean-based aggregation function to model social aggregation for user potential factors in social spaces.

CWYGNN-μ: This is expressed as user attention. The perception of intersection in aggregated opinions is that CWYGNN's user attention μ is disregarded when interacting with one another. This variant employs a mean-based aggregation function to model the potential factors of an item, aiming to achieve the objective of user aggregation.

CWYGNN-head: The multi-head self-attention mechanism is a way of representing users and items as high-dimensional vectors. This mechanism projects them into a shared low-dimensional space, enabling a more precise comparison of their similarities. This approach can significantly enhance the accuracy and efficiency of recommender systems.

Based on the findings presented in Figure 5, using different attention mechanisms on CWYGNN has resulted in varying outcomes.

•Different users have varying contributions to the latent factors in interactive projects such as purchase history, and some users are more significant than others in learning about these factors. The CWYGNN model uses two separate attention mechanisms for evaluation. Experimental findings suggest that CWYGNN-α, CWYGNN-β, and CWYGNN-μ perform worse than CWYGNN-head. This demonstrates that the self-attention mechanism with long heads is more effective than the attention mechanism that was previously used.

•In conclusion, integrating the long-head attention mechanism in CWYGNN significantly enhances the performance of recommendations.

The impact of embedding size. This section examines how the size of item embedding h and opinion embedding p affect the model's performance.

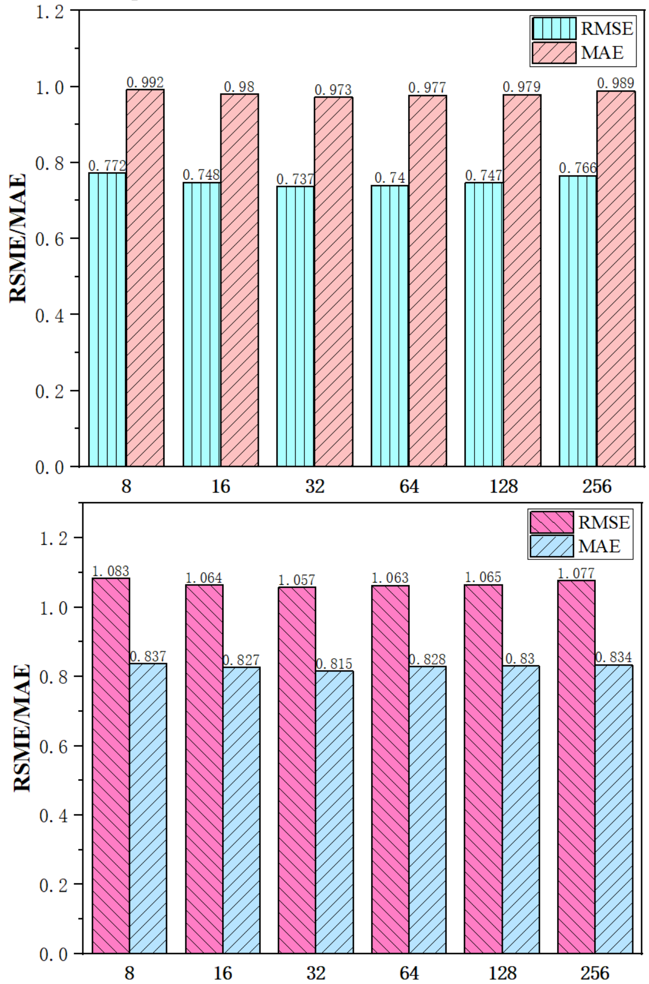

Fig. 6 Impact of embedding size on the ciao and epinions datasets.

The results shown in Figure 6 demonstrate the impact of the embedding length on the Ciao and Epinions datasets. The experiment reveals that increasing the embedding length initially improves the model's performance, but the performance reaches a plateau beyond a certain length. A significant improvement in the model's performance is observed when the range of embedding length varies between 8 and 32. This suggests that increasing the embedding length within this range helps improve the model's ability to represent the data and, therefore, the model's performance. However, when the embedding length is too large, such as up to 256, the performance of the CWYGNN model starts to degrade. This may be because a sizeable embedding length increases the complexity of the model, making it difficult for the model to learn compelling features from the data, thus reducing the model's performance. Based on these observations, we conclude that while a larger embedding size can enhance the model's representation, an excessively long embedding length may increase the model's complexity and diminish its performance. Therefore, we must find an optimal embedding length that balances the model's ability to represent the data with the control of model complexity. By adjusting the embedding length, we can control the model complexity while ensuring the model performance, thus achieving better experimental results.

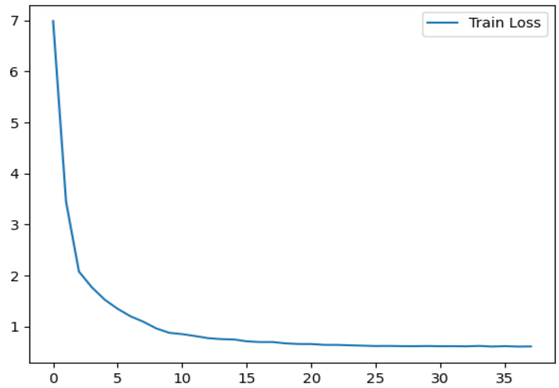

Fig. 7 Model performance evaluation.

In Figure 7, we can see the loss function curve of the CWYGNN model during the training process. The curve for the training set shows a decreasing trend as the number of iterations increases, indicating that the model is continuously learning and optimizing to reduce prediction errors. The model adjusts its parameters and weights with each iteration to minimize the loss function, gradually improving its prediction accuracy. It's worth noting that after the 37th iteration, the training set's Root Mean Square Error (RMSE) reaches a relatively low level. RMSE is a commonly used evaluation metric to measure the difference between predicted and actual values of the model. When the RMSE is low, the model's predictions are closer to the actual situation, which means the model's performance is better. In this case, the RMSE of the training set reaches a low level after the 37th iteration, indicating that the model has converged to a better state. From the training loss curve shown in Fig. 7, we can conclude that the CWYGNN model gradually optimizes and reduces prediction error during the training process and finally reaches a better state after the 37th iteration, which shows a lower RMSE in the training set. According to the study, the CWYGNN model performs exceptionally well during the training process. It learns and optimizes effectively and significantly reduces the prediction error. This helps to improve the accuracy and reliability of the prediction, which is of utmost importance. The model's superior performance provides a solid foundation for selecting and adjusting it in real-life scenarios.

IV. Related work

In this section, we present a summary of our research on social recommendation, deep neural network techniques, and state-of-the-art graph neural networks.

In recent years, recommendation systems have been shifting towards algorithms based on deep learning. This article focuses on social recommendation systems, which emphasize the frequent interactions between users to help them filter out irrelevant information by collecting and analyzing data. As a result, social connections have a positive impact on improving the effectiveness of recommender systems.

The Internet is increasing today, and deep neural network technology has significantly advanced image data processing. One of the most significant achievements of deep neural networks is in image classification tasks. These network structures are known as graph neural networks (GNNs). The main idea behind GNN is to incorporate feature information from local graph neighborhoods by utilizing neural networks. As this process progresses, the node's data is spread throughout the transformed and accumulated graph.

As a result, GNN is excellent at integrating node information and graph topology, and it has proven to be highly effective in representation learning. The emergence of deep neural network models has profoundly impacted various fields, such as image recognition, natural language processing, speech recognition, and more. Other researchers have proposed neural collaborative filtering frameworks like NeuMF to uncover nonlinear interactions between users and projects. This framework can enhance the ability of recommender systems to understand users' interests and needs, resulting in more personalized recommendation outcomes.

Integrating deep neural networks into social recommendation systems has gained momentum recently. For example, NSCR uses the NeuMF model to facilitate cross-domain social recommendations. This involves suggesting items from the information domain to potential users of social networks. The methodology proposes a ranking system for neural social collaboration. The NeuroMF model uses neural networks to represent and match the features of users and items. This leads to more precise recommendations by considering user similarities and differences and the degree of association between users and items.

SMRMNRL employs the perspective of learning multimodal heterogeneous networks for ranking methods to illustrate the evolution of socially aware movie recommendations in social media. They utilized recurrent neural networks and convolutional neural networks to comprehend the representation of movie text descriptions, among other elements. Subsequently, random walk-based methods were incorporated to integrate multimodal neural networks [19-20]. Through these initiatives, they addressed the challenge of cross-domain social recommendations, which significantly deviate from conventional social recommendation systems.

Two models, DeepSoR and GraphRec, are highly relevant to the CWYGNN model. In their research, the DeepSoR team proposes a novel approach that integrates neural networks of user-social relationships with probability matrix factorization. Firstly, they use pre-trained node embedding techniques to represent users more effectively and better capture their personal information and interests. They then use the k-nearest neighbor algorithm to establish a connection between the user's embedded features and the neural network. This improves the accuracy of recommendations. The GraphRec model also introduces "relational embedding," which refers to the ability to capture nodes and edges in a graph using a vector representation of their relationships. By learning these embeddings, the model can better understand the graph's structure, enhancing its performance when tackling graph relationship classification tasks.

GraphRec is a recommendation technique that utilizes attention mechanisms to accurately and personally recommend items to users. This approach models users' unique interests and preferences by aggregating their social and network information. GraphRec aims to promote communication and interaction within social networks by providing personalized recommendations to users. This technology has presented new opportunities and challenges to the development of social networks.

Graph neural networks (GNNs) [21] have shown impressive capabilities in dealing with graph-structured data, especially in recommendation system tasks where the complex relationships between users and items are considered archetypal graph data. Researchers have developed models based on GNNs, such as SRMGCNN [22-23], which uses GNN techniques to generate graph embedding representations to address the recommendation dilemma. Another model, the graph auto-encoding framework proposed by GCMC, generates latent features for users and items by propagating distinct messages across the user-item graph. This framework can extract valuable features by leveraging node and edge representations through message passing, making it vital for social network analysis and personalized recommendations.

PinSage's [24] random walk graph neural network is also a promising approach for learning node embedding representations in large-scale web graphs, which holds immense potential in extensive graph data analysis and modeling.

Although researchers have made impressive strides in various fields, relatively little research has been conducted regarding social recommendation using graph neural networks [25-26]. According to the literature, this area of research has received limited attention [27] so far. However, social recommendation is a crucial aspect of modern society. Therefore, this paper aims to fill this research gap by proposing a social recommendation method based on graph neural networks.

V. Conclusions and Future Work

Information explosion in the current age has made exchanging information between users increasingly tricky. A recommender system focusing on social interactions is essential in such a scenario. However, existing recommendation models have limitations when handling social information, which challenges developing these systems. We propose a new social recommendation model called CWYGNN to address this issue. This model uses graph neural networks and augmented multi-head self-focused networks to capture social relationships and interest preferences among users. The CWYGNN model adopts a multi-head self-attention mechanism, which better captures the interactions between users and improves the accuracy of the recommendation system. Additionally, the model incorporates a graph convolutional network model that better captures interactions and opinions in the user-item graph, leading to more accurate user recommendations. We conducted experiments on two real-world datasets to validate the effectiveness of the CWYGNN model. The experimental results show that the CWYGNN model outperforms similar methods regarding recommendation accuracy and recall, proving its superiority in social recommendation. Although the CWYGNN model has already achieved good results, many other directions are worth exploring for future work. We plan to delve deeper into the underlying factors of user-to-user dynamics better to understand users' social relationships and interest preferences. Additionally, we plan to incorporate more user-to-user social information in developing our social recommendation system to provide users with more personalized recommendations. The CWYGNN model offers a new perspective on solving problems in social recommendation. We believe that the CWYGNN model will continue to play an essential role in our future work by providing users with a better recommendation experience.

References

[1] Hao Ma, Haixuan Yang, Michael R. Lyu and Irwin King, "SoRec: Social recommendation using probabilistic matrix factorization," *Proceedings of the eleventh ACM conference on Knowledge discovery in data mining*, pp.931-940, 2008.

[2] Gao, Peixin, Hui Miao, John S. Baras, and Jennifer Golbeck, "Star: Semiring trust inference for trust-aware social recommenders," *In Proceedings of the 10th ACM conference on Recommender systems*, pp. 301-308, 2016.

[3] Zhao, Rongmei, Shenggen Ju, Zhongzhi Li, and Binyong Li, "A hierarchical attention recommender system based on cross-domain social networks," *Complexity*, pp. 1-13, 2020.

[4] Duan, Y, Zhang, G, Wang, S, Peng, X, Ziqi, W, Mao, J, Wu, H, Jiang, X and Wang, "CaT-GNN: Enhancing Credit Card Fraud Detection via Causal Temporal Graph Neural Networks," arXiv: 2402.14708, 2024.

[5] Zhao, Hengshuang, Jiaya Jia, and Vladlen Koltun, "Exploring self-attention for image recognition," In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition," arXiv:2004.13621, 2020.

[6] Park, Geondo, Chihye Han, Wonjun Yoon, and Daeshik Kim, "MHSAN: multi-head self-attention network for visual semantic embedding," *In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1518-1526, 2020.

[7] Huang, Gao, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q. Weinberger, "Deep networks with stochastic depth," *In Computer Vision–ECCV 2016: 14th European Conference*, vol. 4, no. 14, pp. 646-661, 2016.

[8] Han, Song, Huizi Mao, and William J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," arXiv:1510.00149, 2015.

[9] Y. Liu, Q. Wang, K. Zhao and Y. Liu, “Real-time defect detection of hot rolling steel bar based on convolution neural network,” Chinese Journal of Scientific Instrument, vol. 42, no. 12, pp. 211-219, 2021.

[10] Du, Shaohui, Zhenghan Chen, Haoyan Wu, Yihong Tang, and YuanQing Li, "Image recommendation algorithm combined with deep neural network designed for social networks," *Complexity*, pp.1-9, 2021.

[11] Minaee, Shervin, Yuri Boykov, Fatih Porikli, Antonio Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos, "Image segmentation using deep learning: A survey," *IEEE transactions on pattern analysis and machine intelligence*, pp. 3523-3542, 2021.
[12] Ma, Hao, Haixuan Yang, Michael R. Lyu, and Irwin King, "Sorec: social recommendation using probabilistic matrix factorization," *In Proceedings of the 17th ACM conference on Information and knowledge management*, pp. 931-940, 2008.
[13] Ma, Hao, Dengyong Zhou, Chao Liu, Michael R. Lyu, and Irwin King, "Recommender systems with social regularization," *In Proceedings of the fourth ACM international conference on Web search and data mining*, pp. 287-296, 2011.
[14] Jamali, Mohsen, and Martin Ester, "A matrix factorization technique with trust propagation for recommendation in social networks," *In Proceedings of the fourth ACM conference on Recommender systems*, pp. 135-142, 2010.
[15] Yang, Bo, Yu Lei, Jiming Liu, and Wenjie Li, "Social collaborative filtering by trust," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no.8, pp. 1633-1647, 2016.
[16] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua, "Neural Collaborative Filtering," *In Proceedings of the 26th international conference on world wide web*, pp.173-182, 2017.
[17] Fan, Wenqi, Qing Li, and Min Cheng, "Deep modeling of social relations for recommendation,"*In Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, pp. 8075–8076, 2018.
[18] Fan, Wenqi, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin, "Graph neural networks for social recommendation," *In The world wide web conference*, pp. 417-426, 2019.
[19] Wang, Cheng, Mathias Niepert, and Hui Li, "Recsys-dan: discriminative adversarial networks for cross-domain recommender systems." *IEEE transactions on neural networks and learning systems*, vol.31, no. 8, pp.2731-2740, 2019.
[20] Halvagal, Manu Srinath, and Friedemann Zenke, "The combination of Hebbian and predictive plasticity learns invariant object representations in deep sensory networks," *Nature Neuroscience* , vol.26, no. 11, pp.1906-1915, 2023.
[21] Xiaodan Zhang, Chengwei Huang, and Yongsheng Chen, "Highway Risk Prediction and Factor Evaluation using Convolutional Neural Networks," *IAENG International Journal of Computer Science*, vol. 49, no.1, pp30-35, 2022.
[22] Liu, Chun, Yuxiang Li, Hong Lin, and Chaojie Zhang, "GNNRec: gated graph neural network for session-based social recommendation model," *Journal of Intelligent Information Systems*, vol. 60, no.1, pp. 137-156, 2023.
[23] Liu, Changwei, Kexin Wang, and Aman Wu, "Management and Monitoring of Multi-Behavior Recommendation Systems Using Graph Convolutional Neural Networks," *International Journal of Foundations of Computer Science*, vol. 33, no.6, pp. 583-601, 2022.
[24] Li, X, Fu, C, Zhao, Z, Zheng, G, Huang, C, Dong, J and Yu, Y, "Dual-Channel Multiplex Graph Neural Networks for Recommendation," arXiv:2403.11624, 2024.
[25] Lu, J and Su, R, "Graph neural network news recommendation based on weight learning and preference decomposition," *Journal of Electronic Imaging*, vol. 33, no.1, pp. 011002-011002, 2024.
[26] Bin Zhang, Dan Yang, and Zhihuang Lin, "Medicine and Disease Association Prediction via Attention-Based Medical Heterogeneous Information Network Representation Learning," *IAENG International Journal of Computer Science*, vol. 49, no.1, pp. 69-78, 2022.
[27] Yu Gao, Yin-Qing Wang, Xiao-Long Kong, Fei Guo, Yao-Yang Jin, and Meng-Yuan Xue, "VSG Multi-Machine Parallel Power Decoupling Strategy Based on Improved Virtual Impedance," *IAENG International Journal of Computer Science*, vol.50, no.4, pp. 1159-1165, 2023.