An Improved Gated Graph Neural Network for Sports Tourism Recommendation: User Embedded Representations and Attention Mechanisms

Shu Yu[1*], Keliang Li[2], Guowei Zhao[3]
[1]School of Winter Olympics, Harbin Sport University, Harbin 150006, China
[2]Office of Academic Affairs, Harbin Sport University, Harbin 150006, China
[3]School of Sports Humanities and Society, Harbin Sport University, Harbin 150006, China
E-mail: 15146618866@163.com
[*]Corresponding Author

Keywords: gated unit, user embedded representation, travel recommendation algorithm, attention mechanism

Received: March 5, 2025

In recent years, as the tourism industry rapidly develops, personalized recommendation systems have become an important tool for improving user experience and satisfaction. However, traditional methods face problems such as insufficient recommendation accuracy and low computational efficiency when dealing with large-scale data and complex user needs. Therefore, a sports tourism recommendation model (TRM) based on improved gated graph neural network and user embedded representation is proposed. The model integrates user behavior data and attraction features, and adds attention mechanism and gated unit to improve recommendation accuracy. The dataset used in the study is publicly available on TripAdvisor, which includes detailed user reviews, ratings, and historical behavior data of tourists. The baseline models used for comparison are graph neural networks and attention-based graph neural networks. The performance of this model is evaluated based on several metrics, including F1 score, accuracy, AUC, and inference speed. The research results indicate that the proposed model achieves the highest F1 score of 0.95 after approximately 150 iterations and an accuracy of 0.98 after approximately 100 iterations. Moreover, the model performs outstandingly in terms of recommendation accuracy, relevance, and computational efficiency, with an AUC value of 0.97, inference speed of 0.02 seconds, and computation time of 45 seconds. The findings denote that the proposed model effectively improves the personalization and computational efficiency of tourism recommendations, and can provide users with more accurate recommendations of tourist attractions.

Povzetek: Prispevek predstavi izboljšan model Gated Graph Neural Network z uporabniškimi vdelanimi predstavitvami in mehanizmom pozornosti za natančnejše priporočanje turističnih destinacij v športnem turizmu.

1 Introduction

With the rapid development of the tourism industry, personalized recommendation systems have gradually become an important tool for improving user experience and enhancing user satisfaction. Traditional tourism recommendation methods often rely on rule-based recommendation systems or simple content-based recommendation methods, which often cannot effectively handle users' complex needs and diverse preferences. Therefore, data-driven personalized recommendation systems have become a hot research topic [1-2]. In recent years, as the machine learning and deep learning technologies advance, graph neural networks (GNNs) have achieved significant results in the application of recommendation systems. GNN can effectively capture the relationships between attractions by processing graph structured data, and extract potential patterns between users and attractions through node aggregation operations. However, existing GNN methods still have some limitations in practical applications, especially when dealing with the temporal nature of user behavior and complex relationships between attractions, which often perform inadequately. Therefore, a sports TRM based on improved gated graph neural network (GGNN) and user embedded representation is proposed. This model improves the GNN model by introducing attention mechanisms (AM) and gated units, enabling the model to fully utilize user behavior data and enhance its computational efficiency and recommendation accuracy. Introducing user embedding representation in personalized recommendation systems can significantly improve the accuracy of recommendations, as it can better capture users' personalized features and preferences. However, this process also brings some computational trade-offs. User embedding representation requires additional parameters to represent the behavior and preferences of each user. This means that during the training process, the model needs to store and update a large number of embedding vectors. As the number of users increases, the demand for storage and computing will also sharply rise, leading to an increase in training time. Embedded representation is usually a high-dimensional vector, and each user has an embedded vector. This leads to significant memory consumption, especially on large-scale datasets. As the size of the dataset expands, storing embedded representations for each user will consume a significant amount of memory space, further increasing hardware requirements. The research aims to provide a more accurate and efficient tourism recommendation system, and provide new ideas for the field of personalized recommendation research. The main research questions include: 1) How can the GGNN model outperform traditional GNN and AGNN models in recommendation accuracy by introducing user embedding representation and AMs. 2) Is there a trade-off in the computational efficiency of GGNN while improving recommendation accuracy. 3) How does this model handle the cold start problem and can it maintain good recommendation performance in situations where data is scarce for new users or new attractions. The main contributions of the research include: proposing a TRM based on improved GGNN, which combines gate units and AM, significantly improving recommendation accuracy and computational efficiency. By introducing user embedding representation and attraction embedding representation, the model can capture users' interests and attraction features more personalized, overcoming the shortcomings of traditional methods in big data and dynamic user needs. An innovative time series modeling method has been proposed to enable recommendation systems to consider the dynamic changes in user preferences, thereby improving the real-time and accuracy of recommendations. The research is mainly divided into five sections. The first section reviews research in related fields, and the second section provides a detailed description of the proposed model structure, including improvements to GGNN, learning methods for user and attraction embeddings, introduction of gating units and AMs, etc. The third section presents the experimental setup and results, comparing the performance of this model with other benchmark models. The fourth section analyzes the advantages, limitations, and future improvement directions of the model. The fifth section summarizes the work of this article and proposes possible directions for future research. The innovation of the research lies in combining user embedded representations with scenic spot features, using gating mechanisms to dynamically adjust the data flow, to better capture the relationship between user preferences and scenic spots.

2 Related works

With the rapid development of the tourism industry, personalized recommendation systems have become an important tool for improving user experience and satisfaction. However, traditional recommendation methods still face problems such as insufficient recommendation accuracy and low computational efficiency when dealing with large-scale data and complex user needs. Rabiu I et al. developed an emotion rating model grounded on long short-term neural networks to address the problems of sparse and imbalanced historical rating data in recommendation systems. This model combined functions to capture emotional biases between user ratings and comments. The research findings indicated that the model could effectively solve the above problems and demonstrated good recommendation performance [3]. Yang Z et al. proposed a method that utilizes unlabeled sample information and location information to raise the accuracy of recommendation algorithms, to address the problem of discarding useful information from unknown entries in the rating matrix in traditional recommendation methods. Meanwhile, to reduce computational complexity, the research team also designed an approximate solution model. The research findings indicated that the method had good predictive performance and exhibited robustness to the diversity of the dataset [4]. Zare A et al. proposed a recommendation system framework grounded on supply chain interaction to address the lack of research on interpersonal relationships in social networks. The system adopted a hybrid method combining artificial neural networks and fuzzy strategies, aiming to recommend similar users to social network users. The research results indicated that this method had good recommendation performance and could effectively match similar individuals [5].

Mohammadi N et al. found that many people are increasingly using online services to meet their needs, but there are many users who lack professional knowledge and cannot express their needs well. The current recommendation system only supports users with precise expression ability. In response to this issue, the research team proposed a powerful recommendation system to meet the demands of various users. The system was capable of collecting user preference information and making information recommendations based on various types of information. The research results indicated that the method model had good effect in both accuracy and scalability, and the accuracy of the model was higher when the dataset capacity was large [6]. Benabbes K et al. believed that as the amount of online information continued to increase, it has exceeded the scope of human processing and effective utilization, resulting in extremely low information utilization rates. A recommendation system can solve this problem. The research team applied existing recommendation models to interpret comment texts for user recommendations. The findings indicated that the model had good performance in interpreting comment texts to obtain user information and make recommendations [7]. Peng B found that although the development of e-commerce has facilitated people, the choice of things can lead to the emergence of information processes, which in turn makes it hard for people to find what they want. In response to this issue, the research team proposed a recommendation system based on fuzzy rough sets and cellular algorithms, which can provide personalized recommendations to users grounded on purchase history, and etc. The results indicated that the recommendation system could provide targeted recommendations to users [8]. Xie X et al. proposed an evaluation model based on snowfall, temperature, and wind speed to enrich the connotation of climate suitability for ice and snow sports and conduct cross regional comparisons. They used meteorological data from 1991 to

2021 to study major ski tourism destinations in China. The results indicated that the overall suitability of different regions has decreased, with significant differentiation and spatial heterogeneity in northern Xinjiang. Snowfall was the main influencing factor [9].

In summary, although recommendation algorithms have been widely applied in multiple fields, existing recommendation system methods have some limitations. Firstly, many traditional rule-based and content-based recommendation methods exhibit poor generalization ability when facing diverse or unfamiliar user behaviors. For example, traditional methods that rely on collaborative filtering cannot effectively provide accurate recommendations for new users or cold start users because they fail to fully utilize users' contextual information. Secondly, many methods fail to effectively capture the temporal characteristics of user behavior. The preferences and behaviors of users change over time, and traditional static models fail to consider this temporal dependence, resulting in a decrease in recommendation performance. Many SOTA methods have low computational efficiency, especially large-scale graph models such as GNN, which require a large amount of computing resources during training and inference. When processing large datasets, there are often computational bottlenecks that cannot meet the needs of real-time applications. Finally, some methods fail to fully focus on learning user embedded representations and lack dynamic adjustments based on user behavior, which leads to recommendation systems failing to accurately capture user preferences. This study proposes a new travel recommendation algorithm by improving GNN and user embedded representation. This model aims to provide personalized tourist attraction recommendations for users by integrating their historical behavior and feature information of attractions, especially for those who have travel needs but have not yet made a clear destination choice. The summary of the above research is shown in Table 1.

Table 1: Summary of model performance indicators

Method	Dataset	Accuracy	F1 score	AUC	Remarks
Rabiu I et al. [3]	Custom Emotion Dataset	0.84	0.71	0.81	Emotion-aware recommendation model
Yang Z et al. [4]	Custom Location Dataset	0.85	0.75	0.80	Combines location info for accuracy
Zare A et al. [5]	Custom Social Dataset	0.89	0.8	0.83	Hybrid ANN & Fuzzy system
Mohammadi N et al. [6]	Custom E-Commerce Dataset	0.90	0.82	0.87	Multi-information recommendation
Benabbes K et al. [7]	Custom Reviews Dataset	0.88	0.79	0.85	Based on text interpretation
Peng B. [8]	E-Commerce Dataset	0.86	0.77	0.81	Fuzzy rough sets & cellular algorithms

3 Sports tourism recommendation

3.1 Tourist attraction recommendation model based on GNN

Tourist attraction recommendation is a service that recommends the most suitable tourist attractions to users grounded on their interests, needs, and preferences through algorithms. Its main purpose is to provide users with personalized and accurate tourist attraction recommendations, helping tourists find the most suitable destination among numerous choices. The tourist attraction recommendation model is shown in Figure 1.

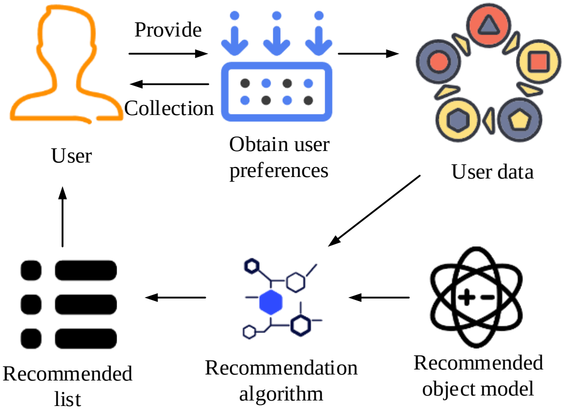


Figure 1: Structure of tourist attraction recommendation model

As shown in Figure 1, the user first provides personal preference data, and the recommendation system obtains the user's preference information based on this data. After obtaining user preferences, the system processes them using recommendation algorithms and generates recommendation results. Subsequently, these recommendation results will be conveyed to the users and further adjustments will be made based on their feedback. Throughout the process, the recommendation system continuously optimizes its recommendation strategy by collecting user preferences and feedback, forming a dynamic adjustment loop to achieve personalized recommendations. The key modules of recommendation methods, user preference models (UPMs), and recommendation result models work together to ensure that the recommendation system can provide users with accurate personalized recommendation services [10]. Through this iterative mechanism, the system can not only satisfy the diverse requirements of users, but also continuously optimize recommendation accuracy and improve user experience. In GNN, each node corresponds to a tourist attraction, and the edges between nodes represent the correlations or relationships between these attractions, such as shared features or user preferences that connect them. In each convolutional layer, nodes aggregate information from their neighboring nodes to update their representations. The expression for feature aggregation is denoted in equation (1).

$$\mathbf{h}_v^{(l+1)} = \sigma\left(\sum_{u \in \mathrm{N}(v)} \frac{1}{c_{vu}} \mathbf{W}^{(l)} \mathbf{h}_u^{(l)} + \mathbf{b}^{(l)} \right) \quad (1)$$

In equation (1), $N(v)$ represents the set of neighboring nodes of node v. c_{vu} is defined as the normalization factor, which is employed to compensate for the discrepancy in the number of neighbours associated with each node. This is achieved by normalizing the information from proximate nodes, thereby ensuring that the influence of each neighbouring node is appropriately balanced. This factor can be defined as the reciprocal of the degree of a node, ensuring that the influence of each neighboring node on the target node is fairly considered. $\mathbf{W}^{(l)}$ means the weight matrix of the l th layer, $\sigma(\cdot)$ means the activation function, and $\mathbf{b}^{(l)}$ means the bias term. The features of neighboring nodes are aggregated into the target node to enhance its contextual information. After feature aggregation, the representation of nodes is updated through the output of the previous layer, as denoted in equation (2) [11].

$$\mathbf{h}_v^{(l+1)} = \mathbf{h}_v^{(l)} + \mathbf{h}_v^{(l+1)} \tag{2}$$

In equation (2), $\mathbf{h}_v^{(l)}$ and $\mathbf{h}_v^{(l+1)}$ represent the node representations before and after the update, respectively. By integrating information from the previous and current layers, it can better capture the complex features of nodes [12-13]. However, simply calculating the similarity between attractions is not enough to generate user recommendations. In the actual recommendation process, the user's preference information is combined with the similarity between tourist attractions. Firstly, the embedded representation of the user will be subjected to similarity calculation with the representations of various attractions to obtain the matching degree between each user and the attractions. Then, by calculating the recommendation score for each attraction, the user's level of interest in the attraction and the similarity between the attractions are considered. The similarity of scenic spots is measured by calculating the inner product between node representations, as expressed in equation (3).

$$\sin(v,u) = \frac{\mathbf{h}_v^{(L)} \cdot \mathbf{h}_u^{(L)}}{\|\mathbf{h}_v^{(L)}\| \|\mathbf{h}_u^{(L)}\|} \tag{3}$$

In equation (3), L denotes the final layer of the GNN, and $\text{sim}(v,u)$ represents the similarity between attraction v and attraction u. Similar tourist attractions are recommended to users by calculating the similarity between attractions. Finally, based on the calculated similarity, the most matching attraction is recommended to the user, as expressed in equation (4).

$$\hat{r}_{uv} = \sum_{u \in \mathrm{M}(v)} \mathbf{h}_u^{(L)} \cdot \sin(v,u) \tag{4}$$

In equation (4), $\hat{r}_{uv}$ represents the recommended rating of attraction v to users, and $\mathrm{M}(v)$ represents all attractions similar to attraction v. By calculating the recommendation score for each attraction, personalized recommendation lists can be generated for users. GNN uses fixed weights to weight and aggregate information from neighboring nodes, which ignores the differences in information contribution between neighboring nodes. In fact, the impact of different neighboring nodes on the target node is different. Some neighboring nodes may provide more effective information, while others may be noise [14-15]. Therefore, introducing AM can help the model focus more on important neighbor nodes when aggregating neighbor node information, automatically assigning a weight to each edge, thereby improving the model's expressive power. The structure of the introduced AM is denoted in Figure 2.

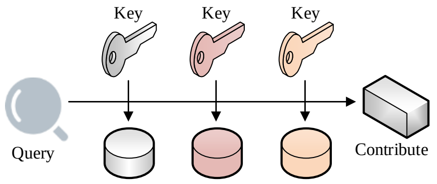


Figure 2: Attention mechanism structure

In Figure 2, the AM is applied within the GNN model to help the system focus on important neighboring attractions during the feature aggregation. The query represents the user's preferences or behavior data, which is used to determine which attractions are more relevant to the user's interests. The keys correspond to the feature representations of the attractions (e.g., location, type, user ratings), while the values represent the attributes or ratings associated with each attraction. The AM calculates the relevance between the query (user's preferences) and the keys (attraction features), assigning higher weights to more relevant attractions. This allows the model to prioritize and aggregate more important features from the neighboring attractions, enhancing the accuracy of the recommendation. The optimized model structure is denoted in Figure 3.

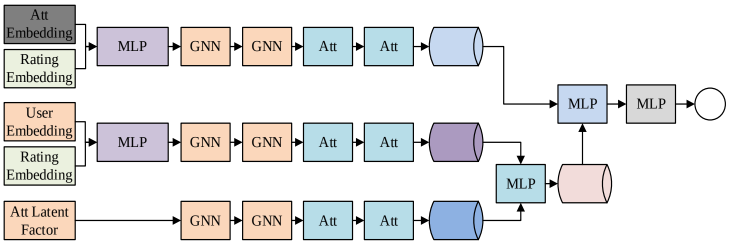


Figure 3: Structure of AGNN model

In Figure 3, the model contains two main modules, namely the UPM and the attraction feature model. In the UPM, the embedded representation of the user is processed together with the rating embedded through a multi-layer perceptron to generate the user's preference information. Then, this information is concatenated with the embedded of scenic spots and potential factors of scenic spots, and input into a multi-layer perceptron to further generate rating predictions. The scenic spot feature model includes multiple attention networks and GNN layers, which are used to process the feature information and graph structure information of scenic spots, respectively, and capture the correlations between scenic spots through AMs and GNN [16]. Finally, through multiple layers of perceptron, the outputs of the UPM and the attraction feature model are combined to generate the final rating prediction.

3.2 Tourism recommendation algorithm based on improved GGNN and user embedded representation

Although the proposed TRM based on GNN can complete most recommendation tasks, the user click sequence is temporal, so inaccurate recommendations may also occur. A TRM based on an improved GGNN and user embedded representation is proposed [17-18]. This model is inspired by recurrent neural network (RNN), which adds gated units to GNN. The time unfolding structure of RNN is shown in Figure 4.

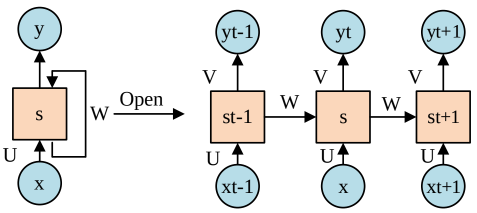


Figure 4: Time evolution of RNN model

In Figure 4, each block represents a moment in the neural network where the input, state, and output of the network interact with each other. At a certain moment, the input value is weighted and inputted into the current state, and the output is calculated through a neural network. Subsequently, the state vector will be passed on to the next moment, and together with the new input, a new state and output will be calculated. This process iterates continuously in time-series data until all time steps are processed, allowing the RNN to capture temporal dependencies when processing sequential data. In this study, the improved GGNN model proposed was inspired by the time series modeling ability of RNNs. However, unlike traditional RNNs, GGNN combines the characteristics of GNNs, especially in each time step, the model not only propagated information through sequence data, but also controlled the information flow in the graph structure through gating units. The introduction of gate control units enabled the model to selectively update node states, retained key memories of user preferences and attraction features, while eliminating noise information. Each node in the graph was embedded with a representation based on its historical behavior and features, and dynamically updated at each time step through the gating mechanism in the GNN. This is similar to the time unfolding structure of RNN, but in GNNs, the state of nodes not only depends on their history, but is also influenced by neighboring node information. On the basis of introducing gated units, the embedding is obtained through projection [19-20]. The structure of the model is denoted in Figure 5.

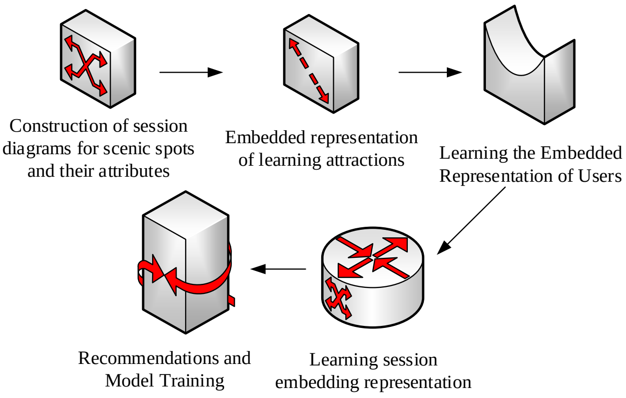

Construction of session diagrams for scenic spots and their attributes
Embedded representation of learning attractions
Learning the Embedded Representation of Users
Recommendations and Model Training
Learning session embedding representation

Figure 5: Framework of the proposed GGNN-based recommendation model

As shown in Figure 5, the five key stages of the recommendation process include the construction of a conversation graph of attractions and attributes, learning of attraction embedding representations, learning of user embedding representations, learning of conversation embeddings, and the final recommendation and model training process. Firstly, constructing a conversation graph of scenic spots and their attributes mainly involves building structured data about each scenic spot and its characteristics. Next, by learning the attraction embedded representation module, the system converts the attributes of the attraction into embedded vectors, allowing the representation of the attraction to be manipulated in low dimensional space [19, 21]. Its expression is shown in equation (5).

$$v_s^t = [x_s^t, 0] \tag{5}$$

In equation (5), $v_{s,i}^t$ is the embedded representation of the s th attraction at time t. $x_{s,i}^t$ is the original input feature of the attraction at time t. At the initial stage of the model, the embedding vector of the scenic spot has no historical information. This zero initialization process is the first step in the model learning process. Over time, by combining GNN and attraction attributes, the embedded representation of attractions will gradually update, capturing the features of attractions and user preferences. The expression for attention score is shown in equation (6).

$$a_{s,i}^t = A_s \cdot [v_{s,i}^{t-1}, \dots, v_{s,i}^0]^T H + b \tag{6}$$

In equation (6), $a_{s,i}^t$ represents the attention score of the s th attraction at time t, which is calculated based on the weight matrix of the attraction and the concatenation of embedded representations from previous time steps.

A_s represents the weight matrix of attraction s, $[v_{s,i}^{t-1},\ldots,v_{s,i}^{0}]$ is the embedding representation, which represents the embedding history of the s th attraction in the past 1 time steps. These embedding representations capture the temporal changes of attraction features. H is the hidden state, and b is the bias term. The equation for updating the door is denoted in equation (7).

$$r_{s,i}^{t} = \tanh(W_r a_{s,i}^{t} + U_r v_{s,i}^{t-1}) \quad (7)$$

In equation (7), $r_{s,i}^{t}$ is the update gate of the scenic spot s at time t, W_r and U_r are the learned weight matrices, $a_{s,i}^{t}$ is the attention score of the scenic spot, and $v_{s,i}^{t-1}$ is the embedded representation of the previous time [22]. Then, the system enters the stage of learning user embedded representations. In this process, the system will learn the user's embedded vector based on their behavior data, capturing their interests and preferences. The control mechanism enables the embedded representation of scenic spots to be flexibly adjusted according to changes in user behavior, rather than relying solely on the static structure of the graph. This improvement enables the model to capture temporal dependencies, that is, changes in user interest in scenic spots, and provide more personalized and accurate recommendations. Among them, the initial embedded representation of the user is shown in equation (8).

$$h_i^0 = (W \cdot \mathrm{Aggreneighbors}(x_k \in N(i))) + b \quad (8)$$

In equation (8), h_i^0 is the initial embedded representation of user i, W is the learned weight matrix, $\mathrm{Aggreneighbors}(x_k \in N(i))$ means the feature vector set of attractions adjacent to user i, and b is the bias term. Specifically, it includes the historical behavior of users and attractions, such as clicks, visits, ratings, etc. These behaviors reflect users' interests and preferences for scenic spots. The attention score is shown in equation (9).

$$y_{ij}^{k} = W_r^2 \cdot [x_i \square x_k \square \Phi_k] + b_1 + b_2 \quad (9)$$

In equation (9), y_{ij}^{k} represents the attention score of attraction k to user i, W_r^2 is the learned weight matrix, x_i denotes the embedded vector of user i, x_k denotes the embedded vector of attraction k, Φ_k is an additional feature representing attraction k, which typically includes additional information such as attraction type, location, rating history, and user preferences, and b is the bias term. Next, the learning session embedded representation module will combine the embedded representations of users and attractions to learn the embeddeds for each session, which helps to understand the needs of users in different conversations [23]. Ultimately, these embedded representations are used to generate recommendation results and model training. In this part, the system generates personalized recommendations based on previously learned representations and trains and optimizes the model. In the "User Embedding Learning" and "Scenic Spot Embedding Learning" stages in Figure 5, the user and scenic spot embeddings obtained separately will be used as inputs along with the session embeddings for processing by the user embedding layer and scenic spot embedding layer in Figure 6. The relationship between users and scenic spots is finely modeled through the information propagation mechanism in GNNs. The final model structure diagram is shown in Figure 6.

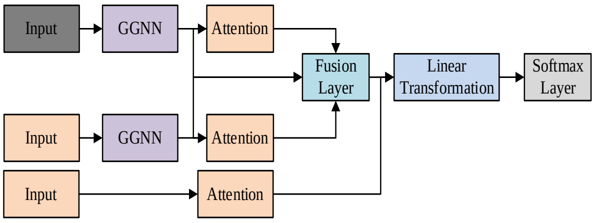


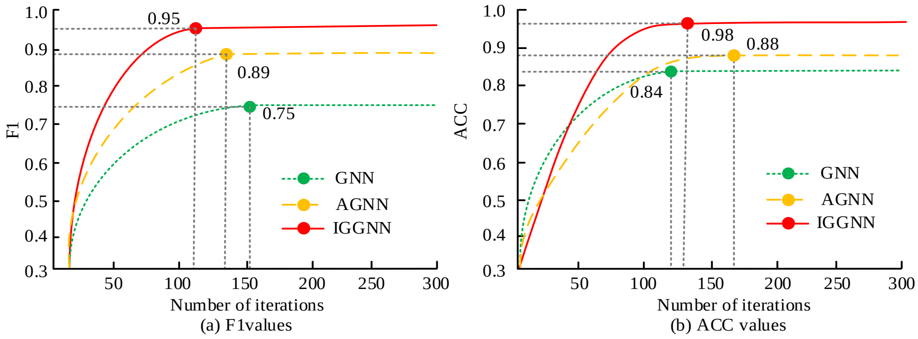
Figure 6: GGNN model architecture diagram

As shown in Figure 6, firstly, the system inputs scenic spot vectors, which are normalized by the Softmax layer to generate probability distributions. Next, these results are connected to vectors from other parts through linear transformation to obtain higher-level feature representations. Then, these input vectors are merged through a fusion layer to generate a comprehensive feature representation that helps the system integrate multiple input information. Next, these data will be input into multiple attention networks, which learn to focus on the important parts of each vector. Next, the system uses a GGNN to update the attraction vector and generate a new representation, while updating the user vector representation through the GNN. Ultimately, through these processed vectors, the system is able to generate personalized recommendation results.

4 Recommendation model performance analysis

4.1 Performance of tourism recommendation algorithm based on improved GGNN and user embedded representation

The study was conducted using a Windows 10 operating system running on a desktop computer with 16G of RAM, an Intel(R) Core(TM) i5-12600KF CPU, and an NVIDIA GeForce RTX 4090D GPU. The study used TripAdvisor's publicly available dataset, which includes travel review data from 500000 users worldwide, covering approximately 10000 tourist attractions. Each attraction was rated on a scale of 1 to 5 based on user feedback. The dataset has high sparsity, with a rating matrix sparsity of 95%, which means that most users did not rate all attractions. Each user comment was accompanied by a timestamp, which records the time of the comment, providing the model with time-series data of user behavior and capturing the temporal changes in user interests. The dataset provided detailed basic information about different locations, such as the names, locations, and types of attractions, as well as the types of restaurants and hotels. In addition to comments and ratings, the dataset also included users' historical behavior data, such as the attractions visited, pages browsed, and favorite destinations. To ensure the best performance of the model on a given dataset, a grid search method was used to optimize hyperparameters. A comprehensive search was conducted for each important hyperparameter to find the most suitable configuration. The parameter settings of the model are as follows: learning rate was 0.001, optimizer used Adam optimizer (default β 1=0.9 and β 2=0.999), batch size was set to 64, training rounds were 150, L2 regularization weight decay was 0.0005, activation function was ReLU, loss function was cross entropy loss, initial embedding dimension was 128, and gradient clipping was set to a maximum gradient of 5. The study chose GNN and AGNN as the comparison models, and the findings are indicated in Figure 7.

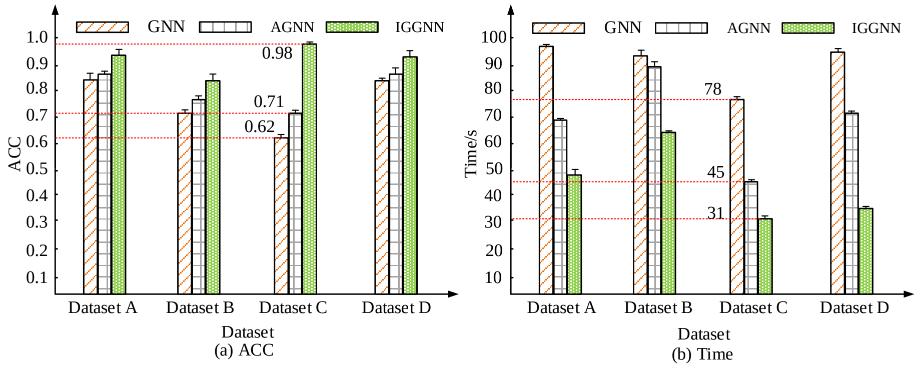


(a) F1values

(b) ACC values

Figure 7: Comparison of F1 and ACC of various models

Figures 7 (a) and 7 (b) show the relationship between the iteration times and F1 score and the relationship between the iteration times and accuracy (ACC), respectively. According to Figure 7 (a), IGGNN reached its highest F1 score of 0.95 after approximately 150 iterations, while AGNN and GNN lagged behind with F1 scores of 0.89 and 0.75. The reason for the superior performance of IGGNN is its more advanced model architecture and more effective regularization techniques, which enable it to better capture important features in tasks. In Figure 7 (b), the IGGNN model performed the best, achieving an ACC of 0.98 after approximately 100 iterations. AGNN and GNN lagged behind with ACC of 0.88 and 0.84, respectively. This model performed well in recommendation ACC, relevance, and computational efficiency, demonstrating more significant advantages than AGNN and GNN. The superiority of IGGNN lied in its more advanced architecture, which effectively combines gating units and AMs. Although AGNN also included these features, IGGNN improved these mechanisms by dynamically adjusting the information flow, allowing the model to focus more on relevant features from past user behavior and suppress irrelevant or outdated data. In addition, the AM in IGGNN helped prioritize the most relevant attractions, ensuring that the model can adapt to changes in user preferences. This dynamic adjustment made IGGNN more advantageous than standard GNN in capturing complex temporal dependencies and generating personalized recommendations. The performance of each model on different types of datasets was analyzed, and the findings are indicated in Figure 8.



(a) ACC

(b) Time

Figure 8: Comparison of ACC and training time of various models on different datasets

Figures 8 (a) and 8 (b) show the ACC and the computation time of three models on different datasets. In Figure 8 (a), the IGGNN model performed the best, with higher ACC than GNN and AGNN on all datasets. On DatasetC, the ACC of IGGNN reached 0.98, significantly better than AGNN's 0.71 and GNN's about 0.62. In addition, IGGNN also maintained high ACC on DatasetA and DatasetD, outperforming other models. As shown in Figure 8 (b), GNN had the longest computation time, with a computation time of nearly 100 seconds on DatasetA, while AGNN and IGGNN reduced computation times to approximately 70 seconds and 45 seconds, respectively. On DatasetC, IGGNN had the lowest computation time, only 31 seconds, which was significantly reduced compared to GNN. The comprehensive performance of each model was analyzed, and the findings are denoted in Table 2.

Table 2: Comprehensive performance analysis table

Model	IGGNN	AGNN	GNN
F1	0.95	0.89	0.75
ACC	0.98	0.88	0.84
Computing time/s	45	70	100
Iterations	150	120	180
Memory usage /MB	550	500	600
AUC	0.97	0.92	0.85
Inference speed /s	0.02	0.03	0.05

According to Table 2, the F1 score of the IGGNN model was 0.95, with an ACC of 0.98, indicating the best performance. IGGNN significantly outperformed other models in both F1 score and ACC, indicating its clear advantages in capturing important features and improving classification performance. The computation time of IGGNN was 45 seconds, significantly lower than AGNN and GNN, indicating that IGGNN also has higher training efficiency. IGGNN required 150 iterations, while AGNN and GNN required 120 and 180 iterations respectively. Although IGGNN had slightly more iterations, it still dominated in other metrics. The AUC value measured the classification performance of the model at different thresholds. The AUC value of IGGNN was 0.97, which was much higher than AGNN's 0.92 and GNN's 0.85, indicating that IGGNN has significant advantages in overall classification performance. The inference speed of IGGNN was the fastest, only 0.02 seconds, while AGNN and GNN were 0.03 seconds and 0.05 seconds respectively, which means that IGGNN can make predictions more quickly in practical applications. In summary, the IGGNN model performs outstandingly in multiple key indicators, with overall performance superior to AGNN and GNN, especially in terms of classification ACC, computational efficiency, and inference speed. The ablation experiment analysis was conducted on the model, and the results are shown in Table 3.

Table 3: Ablation experiment table

Removed component	F1 score	ACC	AUC	Training time (s)	Inference time (s)
Original GGNN model	0.95	0.98	0.97	45	0.02
Remove gated units	0.9	0.94	0.92	60	0.03
Remove attention mechanism	0.91	0.95	0.93	55	0.02
Remove gated units and attention mechanism	0.85	0.9	0.88	70	0.05

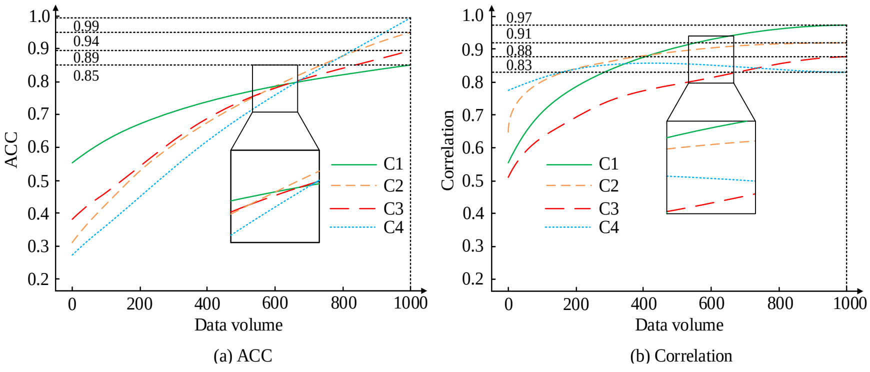
According to Table 3, the original GGNN model showed the highest performance, with an F1 score of 0.95, ACC of 0.98, and AUC of 0.97. When the gating unit was removed, the performance of the model decreased, with F1 score dropping to 0.90, ACC dropping to 0.94, and AUC dropping to 0.92. The training time has also been increased to 60 seconds, indicating that removing the gating unit would affect learning efficiency. The removal of AM could also lead to a decrease in performance, with an F1 score of 0.91 and an AUC of 0.93. Finally, when both the gating unit and AM were removed, the performance of the model significantly decreased, with an F1 score of 0.85, ACC of 0.90, and AUC of 0.88. This indicates that these two components are crucial to the performance of the GGNN model, and their absence can lead to a significant decrease in ACC and effectiveness. The training time has been increased to 70 seconds, which further emphasized the negative impact on learning efficiency when both components were removed.

Compared to AGNN, the GGNN model demonstrated a noticeably faster convergence during training, achieving optimal ACC and F1 score within approximately 100–150 iterations, whereas AGNN required a longer period to reach its peak performance. This trend can be attributed to two main architectural improvements in GGNN: the integration of gated units and user embedded representation. The gated mechanism allowed the model to selectively control the flow of information at each layer, filtering out irrelevant or noisy features and focusing on the most informative inputs. This selective updating process enhanced the model's ability to learn key patterns more efficiently, leading to faster convergence. Additionally, the incorporation of user embedded representations at an early stage gave the model a more expressive initialization of user preferences. As a result, the model could better align user interests with attraction features during the early training phase, accelerating the matching process. In contrast, AGNN lacked the temporal memory and fine-grained control offered by gated units, which results in a slower and more general pattern discovery process. Furthermore, AMs alone, as used in AGNN, were more dependent on large-scale feature interactions to generate meaningful weights, often requiring more training epochs to stabilize.

4.2 Analysis of simulation results of tourism recommendation model

To identify the effect of the model, four different users were chosen for simulation analysis based on their information, and recommended travel destinations that they may be interested in. C1 was an active user with frequent interactions, typically with multiple attractions. The behavior pattern of C2 was relatively clear and stable, but the interaction frequency was slightly lower than C1.

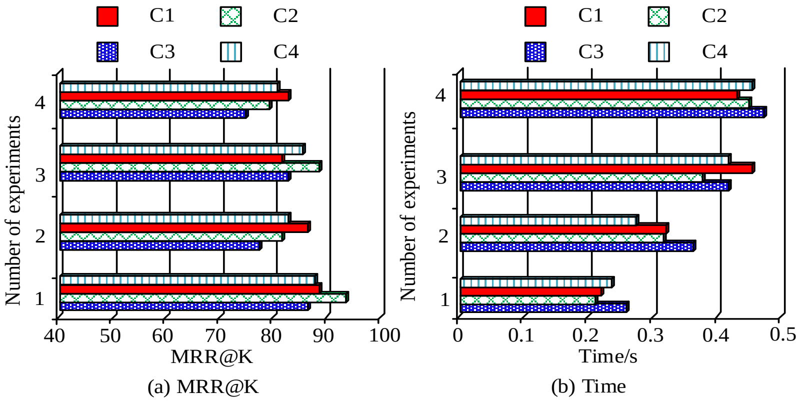
Although the system was able to capture their preferences, less interaction resulted in a relatively slow learning process for the model, especially when the amount of data was small. C3 users had a lower frequency of interaction and exhibited a certain degree of uncertainty or randomness in their behavior. Due to the lack of sufficient behavioral data, the model struggled to accurately capture its preferences, especially in small datasets. C4 was a cold start user with almost no historical behavior data or interaction with attractions. Due to the lack of sufficient interactive information, recommendation systems were unable to effectively provide accurate personalized recommendations, especially when the data volume was small. The results are shown in Figure 9.



(a) ACC (b) Correlation

Figure 9: The recommendation ACC of IGGNN model for different customers

Figure 9 (a) shows the recommendation ACC of the IGGNN model for different customers, and Figure 9 (b) shows the recommendation correlation of the IGGNN model for different customers. in Figure 9 (a), C1 showed an almost linear increase in ACC with the rising of data volume, and the ACC approached 0.95 when the data volume reached 600. In contrast, the performance of C4 was weaker. Despite the increase in data volume, the improvement in ACC was still relatively slow, even reaching only around 0.6 when the data volume was large. However, the overall ACC was above 0.85. From Figure 9 (b), the relevance to C1 rapidly increased when the data volume reached around 500 and stabilized at around 0.97. The relevance performance of C4's recommended content was poor. Although the relevance increased with the increase of data volume, its growth rate was slow and could only reach 0.83 in the end. When the data volume reached 600, the correlation between C2 and C3 stabilized at 0.91 and 0.88, respectively. The findings denoted that the proposed model had excellent recommendation performance. The study selected Mean Reciprocal Ranks (MRR@K) as an indicator. MRR@K measured the location of the scenic spots in the recommended list that best match the user's preferences. The higher the value of MRR@K, the higher the recommended attractions, and the stronger the relevance of the recommendation results to the user. The results are shown in Figure 10.



(a) MRR@K (b) Time

Figure 10: Analysis of MRR@K and recommended time under different experimental times

Figure 10 (a) shows the MRR@K under different experimental times, and Figure 10 (b) shows the recommended time under different experimental times. In Figure 10 (a), for C1, the MRR@K value was significantly higher than other users, staying in the range of 80-90, indicating its efficiency and ACC in recommendation tasks. Relatively speaking, the recommendation performance for C4 was poor, the majority of the MRR@K values were concentrated below 50, for C2 and C3, the MRR@K performance was between C1 and C4. From Figure 10 (b), the time for each recommendation to C4 was the shortest, about 0.2 seconds, indicating that the

C4 model may have high computational efficiency. In contrast, the recommendation time for C1 was the longest, close to 0.4 seconds. The recommendation time of C2 and C3 was between C1 and C4, showing a relatively balanced computational efficiency and ACC. The experimental outcomes showed that the proposed model had excellent effectiveness for all users. The simulation analysis of the model was conducted, and the outcomes are denoted in Table 4.

Table 4: Simulation analysis results of tourism recommendation model

Consumer	ACC	Correlation	MRR@K	Recommended time/s
C1	0.95	0.97	85	0.40
C2	0.9	0.91	78	0.35
C3	0.88	0.88	72	0.31
C4	0.6	0.83	48	0.22
Consumer	The number of experiments	Memory usage/MB	Computing power	/
C1	5	550	0.87	/
C2	5	530	0.83	/
C3	5	540	0.75	/
C4	5	520	0.65	/

In Table 4, computing power represents the CPU/GPU computing power consumed by the model during the training or inference phase, usually indirectly expressed through computation time (in seconds). Higher computing power means that the model requires more time to process data and perform computational tasks. Therefore, computing power is closely related to the running efficiency of the model, which can be measured by the training time and inference time of the model. According to Table 4, C1 had the highest ACC rate, reaching 0.95, indicating the strongest ACC in their recommendation. C4 had a significantly lower ACC rate, only 0.6, indicating poor recommendation performance. The correlation of C1 was 0.97, which performed very well, while the correlation of C4 was 0.83, which was relatively low, indicating that the model failed to effectively capture C4's interests and preferences. C1's MRR@K value was 85, which is greatly higher than other customers, indicating that their recommended results usually rank higher. In contrast, the MRR@K value of C4 was only 48, indicating a low recommendation effect. C4 had the shortest recommendation time, only 0.22 seconds, indicating that its model has high computational efficiency, while C1 had a longer recommendation time, reaching 0.4 seconds. C1 has the strongest computing power at 0.87, performing the best, while C4 had the worst computing power at only 0.65. The findings denoted that the proposed model had excellent performance.

5 Discussion

The IGGNN model proposed in the study significantly outperformed existing GNN and AGNN models in multiple performance metrics, particularly in ACC, F1 score, and AUC value. IGGNN achieved an ACC of 0.98 and an F1 score of 0.95 on the TripAdvisor dataset, while GNN and AGNN performed significantly worse, with accuracies of 0.84 and 0.88, respectively. This improvement can be attributed to the user embedding representation, temporal modeling, and AM introduced in the IGGNN model, which effectively enhance the personalization and ACC of recommendations, and have stronger recommendation performance compared to other methods in existing literature. However, there are also some issues with using IGGNN. Although IGGNN performed well in ACC, its computational complexity was relatively high, especially in the training and inference stages, requiring more computing resources, especially on large datasets, which is similar to the research results of Liu C et al. [24]. Therefore, IGGNN may not be suitable for real-time applications with limited resources. In addition, the interpretability of IGGNN was relatively poor compared to some simple recommendation models. Although AMs were introduced to help the model focus on important features, the complexity of its internal structure still made the decision-making process of the model difficult for users to understand. The performance differences of the model on different datasets were analyzed. IGGNN performed well on the TripAdvisor dataset, but its performance was slightly inferior on other datasets, with an ACC rate dropping to 0.71. This indicated that IGGNN relied heavily on datasets and might not be able to fully utilize its advantages when dealing with datasets with limited data or simple features. Therefore, future research can consider further optimizing the IGGNN model on datasets from different fields to improve its universality. Although IGGNN performs well in most experiments, there are still some limitations. For example, the issue of cold start has not been fully addressed in research. When there is insufficient behavior data for new users or new attractions, the recommendation performance of the model will decrease. Future research can combine transfer learning or reinforcement learning techniques to alleviate this problem. In summary, the IGGNN model proposed in the study has made significant progress in improving recommendation ACC and computational efficiency, but further optimization is still needed in terms of the model's computational efficiency, interpretability, and response to cold start problems.

The IGGNN model proposed in the study has strong deployability in real-world tourism applications. Firstly, the model can process user behavior data in real-time, dynamically adjust recommended content, and ensure the ACC and real-time nature of personalized recommendations. Secondly, IGGNN can support large-scale user and attraction data through efficient computation of embedded representations and GNNs. Although the training phase requires a lot of computing resources, the inference speed is fast and meets real-time recommendation requirements. In addition, the model can provide diversified recommendations to avoid user recommendation fatigue. Finally, the IGGNN model can be regularly updated through incremental learning to maintain continuous performance optimization and adapt to new attractions and changing user needs. In practical deployment, combined with reinforcement learning and other techniques, the model can continuously optimize itself during the application process, improving recommendation effectiveness.

6 Conclusion

A TRM based on IGGNN and user embedded representation was proposed to solve the challenges of traditional tourism recommendation systems in terms of ACC, timeliness, and computational efficiency. By introducing gated units and AMs, IGGNN could effectively integrate user behavior data and attraction features, significantly improving recommendation ACC and system efficiency. The findings denoted that the IGGNN model performed well in indicators such as F1 score, ACC, and AUC value. The IGGNN model had the best performance with an F1 score of 0.95 and an ACC of 0.98. IGGNN significantly outperformed other models in both F1 score and ACC, indicating its clear advantages in capturing important features and improving classification performance. The computation time of IGGNN was 45 seconds, significantly lower than AGNN and GNN, indicating that IGGNN also had higher training efficiency. IGGNN required 150 iterations, while AGNN and GNN required 120 and 180 iterations respectively. Although IGGNN had slightly more iterations, it still dominated in other metrics. The AUC value measured the classification performance of the model at different thresholds. The AUC value of IGGNN was 0.97, which was much higher than AGNN's 0.92 and GNN's 0.85, indicating that IGGNN has significant advantages in overall classification performance. In the simulation analysis, the ACC of C1 was as high as 0.95, and the recommended time was 0.4 seconds. The outcomes denoted that the proposed model had excellent performance. Although the model performed well in multiple indicators, there were still certain challenges for low-frequency users and cold start problems, especially in situations where the data volume is small or user behavior is low, the recommendation performance of the model may decrease. Future research can further improve the recommendation ACC and generalization ability of the model by combining more contextual information and reinforcement learning techniques.

References

[1] Si K, Zhou M, Qiao Y, Sun G. 5G Multimedia Precision Marketing Based on the Improved Multisensor Node Collaborative Filtering Recommendation Algorithm. Journal of Sensors, 2021, 21(11):5893-5910. https://doi.org/10.1155/2021/5856140

[2] Cai X, Hu Z, Chen J. A many-objective optimization recommendation algorithm based on knowledge mining. Information Sciences, 2020, 537(23): 146-161. https://doi.org/10.1016/j.ins.2020.05.067

[3] Rabiu I, Salim N, Da'U A, Nasser M. Modeling sentimental bias and temporal dynamics for adaptive deep recommendation system. Expert Systems with Applications, 2022, 191(8): 11862-11876. https://doi.org/10.1016/j.eswa.2021.116262

[4] Yang Z, Zhou F, Yang L, Zhang Q. A new prediction method for recommendation system based on sampling reconstruction of signal on graph. Expert Systems with Applications, 2020, 159(11); 11357-11369. https://doi.org/10.1016/j.eswa.2020.113587

[5] Zare A, Motadel M R, Jalali A. A Hybrid Recommendation System Based on the Supply Chain in Social Networks. Journal of Web Engineering, 2022, 21(3):633-659. https://doi.org/10.13052/jwe1540-9589.2133

[6] Mohammadi N, Rasoolzadegan A. A two-stage location-sensitive and user preference-aware recommendation system. Expert Systems with Applications, 2022, 191(9):11618-11643. https://doi.org/10.1016/j.eswa.2021.116188

[7] Benabbes K, Housni K, El Mezouary A, Zellou A. Recommendation System Issues, Approaches and Challenges Based on User Reviews. Journal of Web Engineering, 2022, 21(4):1017-1054. https://doi.org/10.13052/jwe1540-9589.2143.

[8] Peng B. Research and Implementation of Electronic Commerce Intelligent Recommendation System Based on the Fuzzy Rough Set and Improved Cellular Algorithm. Mathematical Problems in Engineering, 2021, 202(4):6671-6679. https://doi.org/10.1155/2021/6671219

[9] Xie X, Pang Z, Zhu H. Spatial and Temporal Differences of Climate Suitability of Ice and Snow Sports in Major Ski Tourism Destinations in China. Chinese Geographical Science, 2024, 34(5):967-982. https://doi.org/10.1007/s11769-024-1434-9

[10] Cai B, Zhu X, Qin Y. Parameters optimization of hybrid strategy recommendation based on particle swarm algorithm. Expert Systems with Applications, 2021, 168(12):1143-1158. https://doi.org/10.1016/j.eswa.2020.114388

[11] Chen G, Zeng F, Zhang J, Lu Ting, Shen J, Shu W. An adaptive trust model based on recommendation filtering algorithm for the Internet of Things systems. Computer Networks, 2021, 190(15):107-121. https://doi.org/10.1016/j.comnet.2021.107952

[12] Yang F, Xie H, Li H. Video associated cross-modal recommendation algorithm based on deep learning. Applied Soft Computing, 2019, 82(1):1055-1097. https://doi.org/10.1016/j.asoc.2019.105597

[13] Xu S, Zhuang H, Sun F, Wang S, Wu T, Dong J. Recommendation algorithm of probabilistic matrix factorization based on directed trust. Computers & Electrical Engineering, 2021, 93(2):1072-1093. https://doi.org/10.1016/j.compeleceng.2021.107206

[14] Bin S, Sun G. Matrix Factorization Recommendation Algorithm Based on Multiple Social Relationships. Mathematical Problems in Engineering, 2021, 21(8):1-8. https://doi.org/10.1155/2021/6610645

[15] Gao Y, Liang H, Sun B. Dynamic network intelligent hybrid recommendation algorithm and its application in online shopping platform. Journal of Intelligent and Fuzzy Systems, 2021,40(1):1-13. https://doi.org/10.3233/JIFS-20157

[16] Chen B, Zhu L, Wang D, Cheng J. Research on the Design of Mass Recommendation System Based on Lambda Architecture. Journal of Web Engineering, 2021, 20 (6):1971-1990. https://doi.org/10.13052/jwe1540-9589.20614.

[17] Abbasi-Moud Z, Vahdat-Nejad H, Sadri J. Tourism recommendation system based on semantic clustering and sentiment analysis. Expert Systems with Applications, 2021, 167(8):1143-1153. https://doi.org/10.1016/j.eswa.2020.114324

[18] Ke G, Du H L, Chen Y C. Cross-platform dynamic goods recommendation system. Applied Soft Computing, 2021, 104(1):1072-1081. https://doi.org/10.1016/j.asoc.2021.107213

[19] Gheisari M, Hamidpour H, Liu Y, Saedi P, Raza A, Jalili A, Rokhsati H, Amin R. Data Mining Techniques for Web Mining: A Survey. Artificial Intelligence and Applications, 2023, 1(1): 3-10. https://doi.org/10.1109/ICCIC.2010.5705856.

[20] Luo F, Ranzi G, Kong W, Dong Z. A Personalized Residential Energy Usage Recommendation System Based on Load Monitoring and Collaborative Filtering. IEEE Transactions on Industrial Informatics, 2021, 17(2):1253-1262. https://doi.org/10.1109/TII.2020.2983212.

[21] Chen Z. Research on internet security situation awareness prediction technology based on improved RBF neural network algorithm. Journal of Computational and Cognitive Engineering, 2022, 1(3): 103-108. https://doi.org/10.47852/bonviewJCCE149145205514

[22] Dzemyda G, Sabaliauskas M, Medvedev V. Geometric MDS Performance for Large Data Dimensionality Reduction and Visualization. Informatica, 2022, 33(2):299-320. https://doi.org/10.15388/22-INFOR491

[23] Mehta P, Aggarwal S, Tandon A. The Effect of Topic Modelling on Prediction of Criticality Levels of Software Vulnerabilities. Informatica, 2023, 8(22):283-304. https://doi.org/10.31449/inf.v47i6.3712

[24] Liu C, Zhou K, Zhou L. Infusing external knowledge into user stance detection in social platforms. Journal of Intelligent & Fuzzy Systems, 2024, 46(1):1-16. https://doi.org/10.3233/JIFS-224217