SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V13I6P103
Volume 13 Issue 6, 28-37, June 2026

Original Article

Adaptive Graph Contrastive Multi-Criteria Recommender with Dynamic Preference Evolution

K. Sravanthi[1], T. Archana[2]

[1,2]CSE, Kakatiya University, Warangal, India.

[1]Corresponding Author : sravanthi.k1982@kakatiya.ac.in

Received: 06 March 2026 Revised: 05 April 2026 Accepted: 04 May 2026 Published: 27 June 2026

Abstract *- The increasing amount of online information has resulted in information overload, leading to the need for precise and adaptive recommenders. The existing single-criterion recommenders fail to address the multidimensional nature of user preferences, whereas the existing Multi-Criteria Recommender Systems (MCRS) face the problems of sparsity, limited relational modeling, and static assumptions of user preferences. This paper presents a novel Temporal Graph Contrastive Deep Autoencoder-based Multi-Criteria Recommender System (TC²-GDAE-MCRS) that incorporates criterion-wise deep autoencoders, graph neural networks, contrastive self-supervised learning, temporal modeling of user preferences, and reinforcement learning in a single framework. The proposed model is evaluated using the Yahoo! Movies and TripAdvisor benchmark datasets with rating prediction and top-K ranking metrics. The results show that the proposed system reduces the RMSE from 0.912 to 0.824, MAE from 0.784 to 0.701, and NDCG@10 from 0.736 to 0.812, significantly outperforming the existing state-of-the-art models.*

Keywords *- Multi-Criteria Recommender Systems, Deep Autoencoders, Graph Neural Networks, Contrastive Learning, Temporal Modeling, TC²-GDAE-MCRS.*

1. Introduction

The proliferation of online information due to the increasing popularity of e-commerce, social media, online streaming, and review sites has led to information overload, making it hard for users to find products that suit their interests [1, 2]. To mitigate this, Recommender Systems (RS) have been developed to filter the information overload. However, the existing recommender systems have limitations, including the use of single criterion-based rating systems, which cannot effectively capture the multi-dimensional nature of users' preferences [3]. To address the limitations of the existing recommender systems, the introduction of Multi-Criteria Recommender Systems (MCRS) enables users to rate products based on several factors, including quality, service, and price. Despite the advantages of MCRS, they have limitations, including the problem of sparsity, which affects the performance of the existing collaborative filtering and matrix factorization techniques [1]. Moreover, the existing MCRS considers users' preferences as static, ignoring the relational effects between users, items, and factors [24].

Recent advances in recommender systems have been achieved through the introduction of deep learning, which improves the performance of recommender systems. Deep Autoencoders have been shown to improve the performance of multi-criteria representation learning [3, 6]. However, the existing recommender systems have limitations, including the inability of the existing recommender systems to consider the interaction between users and items. Graph Neural Networks have been shown to improve the performance of recommender systems through the modelling of the interaction between users and items [7, 8]. However, the existing GNN-based recommender systems have limitations, including the inability to handle single-criterion-based systems. Moreover, the existing GNN-based recommender systems lack interpretability [9]. Traditionally, recommender systems have mostly employed collaborative filtering and matrix factorization-based approaches. In the context of the latter, Koren et al. proposed the latent factor model, which proved that matrix factorization can effectively capture the relationship between users and items through the factorization of the sparse rating matrix, hence achieving better accuracy in the prediction results [1]. These approaches rely on the assumption that the preferences of users can be projected onto a shared latent space, and the interaction between users and items can be approximated through linear operations such as the dot product. While such approaches have achieved tremendous success for traditional recommendation problems, they have their own limitations. For instance, they cannot effectively capture the complex interaction patterns between users and items. Moreover, they rely on a single rating value to represent the preference of users, which does not reflect the multi-aspect decision-making behaviour. Another limitation of such approaches is that they fail to generalize to sparse interaction patterns. These limitations have led to the requirement for approaches that can effectively learn the representation based on multi-aspect preferences, structural dependencies, and temporal behaviour [17, 24, 27].

2. Literature Survey

2.1. Traditional Recommender Systems

The classical recommender systems have been based mostly on collaborative filtering and matrix factorization methods that develop latent user and item representations based on historical interaction data. One of them is the latent factor model suggested by Koren et al., who prove that matrix factorization can be useful to model user-item relationships by breaking down the sparse rating matrix into low-dimensional embeddings, thus improving the predictability and scalability of these models [1]. These techniques assume that user preferences lie in a common latent space and the action of a user and a piece can be approximated by the linear operations, such as inner products. Though even the models were known to be relatively successful in the initial tasks of recommendation, they are not intrinsically successful in their ability to model interaction patterns that are inherently nonlinear and complex [4, 5]. In addition, they use one overall rating to capture user preference, which does not manifest multi-aspect decision-making, which is apparent in real-life scenarios. These limitations highlight the need to have models that are able to learn richer representations that can include more dimensions of preferences, structural dependencies, and temporal behavioural patterns [17].

2.2. Multi-Criteria Recommender Systems

Deep learning has greatly impacted the field of recommender systems research, especially when considering the modelling of complex and nonlinear interactions between users and items. Neural Collaborative Filtering replaces the inner product operation in the traditional matrix factorization method with the multilayer neural network model, which learns the high-order interaction functions to achieve better recommendation accuracy [4]. DeepFM also integrates the factorization machines model with the deep neural network model to learn the low-order feature interactions and high-order nonlinear interactions simultaneously [5]. The autoencoder-based models, including the denoising autoencoder and the variational autoencoder, have been broadly adopted to handle the sparse rating matrices by reconstructing the rating matrices [3, 6]. The models have achieved state-of-the-art performance in sparse environments, especially when considering the ability to learn a compact representation from incomplete information. Although the models have achieved the best performance in the sparse environments, the deep learning-based models, including the neural collaborative filtering model, the deepFM model, the autoencoder-based models, and other variants, mainly assume the independent interactions between the user and the item and fail to model the structural relationships between the user and the item. Thus, the models fail to model the high-order collaborative effects from the network structure of the user-item interactions [7, 8]. Furthermore, the models mainly focus on the static environments and fail to consider the dynamic nature of the user behaviour and the evolution of the user preferences over time [24].

2.3. Graph Neural Networks in Recommender Systems

Recently, a new paradigm, namely, graph neural networks, has been proposed, which has shown promise in recommender systems. This is because user interactions with items can be naturally represented using a graph, where the user and item are represented using nodes, while their interactions are represented using edges. The Neural Graph Collaborative Filtering model has been proposed, which incorporates a message passing mechanism to learn higher-order connectivity, thus improving the quality of the embedding significantly [7]. Moreover, a new model, namely, LightGCN, has been proposed, which has removed the non-linear transformation and feature projection matrices, thus reducing the computational cost while retaining state-of-the-art performance [8]. The major advantage of using a graph neural network is that it captures higher-order interactions, such as user-user similarity and item-item similarity, which are not captured using traditional models [28, 29]. Nevertheless, the majority of the existing state-of-the-art models using a graph neural network are applicable to single-criterion ratings, while the interaction graphs are static. This means that these models cannot capture aspect-level preference learning, nor can they capture temporal dynamics, thus failing to capture the evolving nature of user interests, which is critical in next-generation recommendation systems [24].

2.4. Contrastive Learning and Temporal Modelling

Contrastive self-supervised learning has been incorporated into the field of recommender systems to enhance the robustness of the learned representation, especially in environments with sparsity and noise. By learning from the augmented views of the interaction graph through various stochastic perturbation techniques, such as edge dropout and node masking, the contrastive learning model aims to maximize the agreement between the learned representations of the same nodes while minimizing the relationships between unrelated nodes in the graph [18, 19]. Wu et al. have successfully demonstrated the efficacy of the proposed contrastive learning model in enhancing the generalization ability and reducing the cold start issue through the learning of invariant and discriminative representations without the need for any additional labelled data [9]. Concurrently, temporal recommendation models have been proposed to effectively model the dynamic nature of user behaviour and preferences. Sequential models, such as the proposed STAMP model, make use of attention mechanisms to effectively learn the short-term and long-term user interests from the interaction sequences [10]. RNNs and transformer models have also been proposed to effectively model the complex transitions in user behaviour through the incorporation of the temporal dependency learning mechanism [20, 21]. Although significant improvements have been made in the development of contrastive learning models and temporal recommendation models, the proposed models have been developed independently and have yet to be combined with the existing multi-criteria recommendation models and graph-based relational learning models [25, 26]. Furthermore, reinforcement learning-based recommendation, which is capable of optimizing long-term user satisfaction through interaction feedback, is rarely integrated with multi-criteria graph-based deep learning frameworks. As a result, the majority of current models optimize short-term accuracy rather than cumulative recommendation quality [22, 23]. These limitations indicate that the existing methods address different aspects of the recommendation problem in isolation and lack a comprehensive architecture that jointly models multi-criteria semantics, high-order relational dependencies, temporal preference evolution, robust self-supervised representation learning, and long-term ranking optimization [27-29].

3. Proposed Methodology and Methods

3.1. Problem Definition

Let $U = \{u_1, u_2, \dots, u_m\}$ be a set of users, $I = \{i_1, i_2, \dots, i_n\}$ a set of items, and $C = \{c_1, c_2, \dots, c_k\}$a set of rating criteria (e.g., service, quality, price).

Users provide partial multi-criteria ratings represented as a sparse tensor $R \in \mathbb{R}^{m \times n \times k}$, where many entries are missing. Each interaction is associated with a timestamp t, reflecting the temporal evolution of user preferences. The objective of this work is to learn a function $f(u, i, t)$that predicts missing criterion-level ratings and produces accurate top-K item recommendations by jointly modeling multi-criteria semantics, user–item relational structure, and temporal preference dynamics within a unified learning framework.

3.2. Proposed Methodology

The proposed methodology presents an innovative Temporal Graph Deep Autoencoder-based Multi-Criteria Recommender System (TC²-GDAE-MCRS) to address the shortcomings of traditional static multi-criteria approaches and graph-based recommenders. Contrary to the existing state-of-the-art methods, which consider user preferences as static and autonomous in nature, the proposed approach aims to jointly capture multi-criteria semantics, graph relationships, and temporal preference dynamics within a unified learning framework. The complete pipeline involves criterion-wise representation learning via deep autoencoders, followed by relational learning via a graph neural network. To make the model more robust in sparse data settings, contrastive self-supervised learning is added. Next, preference dynamics are captured via sequential encoders, and final recommendations are further optimized via reinforcement learning.

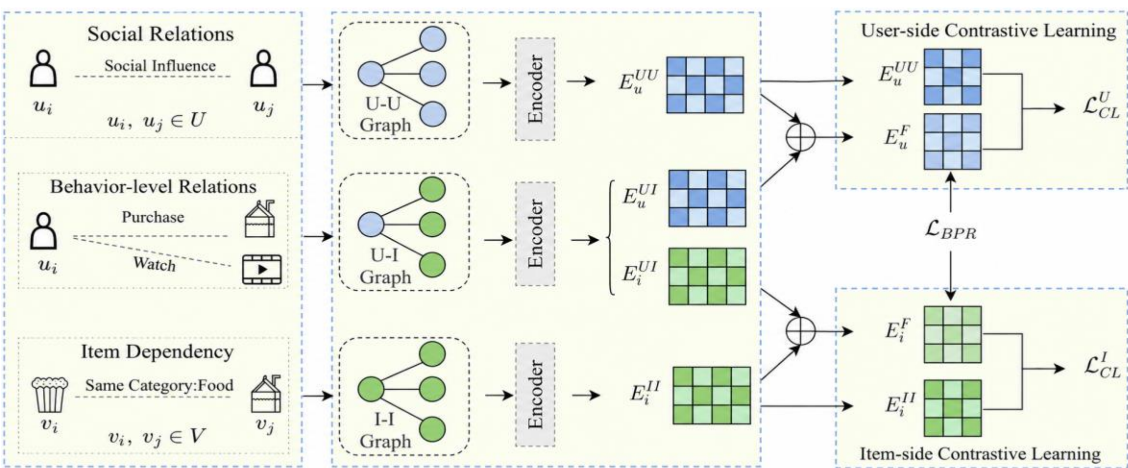

Fig. 1 Overall architecture of the proposed TC²-GDAE-MCRS framework

3.2.1. Unified Optimization Objective

The TC²-GDAE-MCRS framework is trained end-to-end using a unified multi-task objective function defined as:

$$\mathcal{L} = \lambda_{rec}\mathcal{L}_{AE} + \lambda_{con}\mathcal{L}_{CL} + \lambda_{rank}\mathcal{L}_{rank} + \lambda_{RL}\mathcal{L}_{RL} \quad (1)$$

where $\mathcal{L}_{AE}$is the reconstruction loss of criterion-wise autoencoders, $\mathcal{L}_{CL}$is the contrastive self-supervised loss, $\mathcal{L}_{rank}$ is the ranking loss for top-K recommendation, and $\mathcal{L}_{RL}$ represents the reinforcement learning objective. The coefficients λ control the relative importance of each component and are selected via validation.

3.3. Multi-Criteria Representation Learning using Deep Autoencoders

Multi-criteria recommender systems are characterized by multiple aspect-level ratings (e.g., service, quality, value) with each rating having its own set of statistical and semantic properties. To capture these properties, the proposed framework uses criterion-wise deep autoencoders, where each criterion is encoded separately into a latent space.

For a user-item rating of criterion c, the encoder takes the observed rating vector. $r_{u,i}^{c}$ and maps it to a compact latent space:

$$h_{u,i}^{c} = f_{\text{enc}}^{c}\left(r_{u,i}^{c}\right) \tag{2}$$

In this case, $r_{u,i}^{c}$ represents the observed rating (or rating vector) provided by user u to item I for criterion c, $f_{\text{enc}}^{c}(\cdot)$ represents the encoder network with nonlinear activation functions. The decoder reconstructs the input as

$$\hat{r}_{u,i}^{c} = f_{\text{dec}}^{c}\left(h_{u,i}^{c}\right) \tag{3}$$

The autoencoders are trained by minimizing the reconstruction loss with regularization:

$$\mathcal{L}_{AE} = \sum_{u,i,c} \parallel r_{u,i}^{c} - \hat{r}_{u,i}^{c} \parallel^{2} + \lambda \parallel W \parallel_{2}^{2} \tag{4}$$

where $\mathcal{L}_{AE}$: Overall loss function of the autoencoder, $r_{u,i}^{c}$: Actual rating given by user $u$to item $i$for criterion c, $\hat{r}_{u,i}^{c}$: Rating reconstructed by the autoencoder,$\parallel r_{u,i}^{c} - \hat{r}_{u,i}^{c} \parallel^{2}$: Reconstruction error measuring prediction accuracy, W: Trainable weight parameters of the autoencoder, $\parallel W \parallel_{2}^{2}$: L2 regularization term to control model complexity, lambda: This is the regularization coefficient used to control the strength of regularization. This technique essentially filters out noise, deals with missing ratings, and preserves criterion-specific semantics, thus yielding strong latent embeddings.

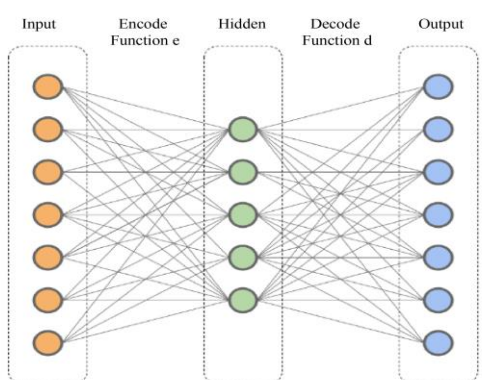

Fig. 2 Architecture of the deep autoencoder-based multi-criteria recommender system

3.4. Graph Neural Network for Relational Learning

In order to represent the interaction between the user, the items, and the context in a more complex way than the ratings, the system uses a heterogeneous interaction graph, in which the entities of the user, the items, and the context are represented as nodes, while the interaction criteria are represented as edges.

A Graph Neural Network is used to aggregate information from this graph. The representation of a node v at layer $(l+1)$ is given by:

$$h_{v}^{(l+1)} = \sigma\left(\sum_{u \in \mathcal{N}(v)} W^{(l)} h_{u}^{(l)}\right) \tag{5}$$

where $\mathcal{N}(v)$ represents the neighbourhood of node v, and $W^{(l)}$ These are learnable parameters. By multi-layer message passing, the GNN learns the high-order dependencies like similarity between users, similarity between items, and interactions between criteria, which are not explicitly represented in the rating matrices.

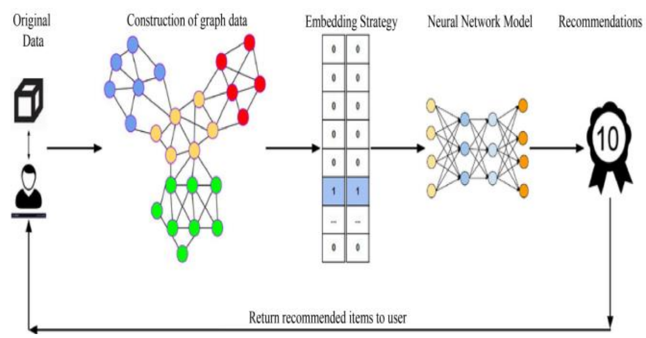

Fig. 3 Graph-Enhanced Deep Autoencoder-Based Multi-Criteria Recommender System (GDAE-MCRS)

3.5. Contrastive Graph Representation Learning

Though GNNs are an improvement in relational learning, there could still be instability in the embeddings if there is sparse interaction data. To overcome this problem, the proposed framework combines contrastive self-supervised learning to improve the robustness of the embeddings.

Two different graph views are created using stochastic perturbations, such as edge dropout or node masking. Let z_{v} and z_{v}^{+} be the embeddings of the same node in two graph views. The contrastive loss function is given by:

$$\mathcal{L}_{CL} = -\log \frac{\exp(\text{sim}(z_{v}, z_{v}^{+})/\tau)}{\sum_{j} \exp(\text{sim}(z_{v}, z_{j}^{-})/\tau)} \tag{6}$$

where τ is a temperature parameter and $\text{sim}(\cdot)$is a similarity function. This loss encourages consistent embeddings across views while separating unrelated nodes, significantly improving generalization and cold-start performance.

3.6. Temporal Preference Evolution Modelling

User preferences are dynamic and change over time. To model this, the framework uses temporal preference modelling with sequential neural networks. With a sequence of user interactions over time, the dynamic preference state is modelled as follows:

$$s_u^t = \mathrm{GRU}(s_u^{t-1}, z_u^t) \quad (7)$$

where z_u^t is the representation of the fused embedding at time t. This temporal encoder helps the model capture short-term interests and long-term preferences of the user, hence making recommendations that are dynamic based on user behaviour.

3.7. Feature Fusion and Attention-Based Aggregation

The final user-item representation is achieved by concatenating graph-based embeddings, criterion-wise autoencoder features, and temporal preference states:

$$z_{u,i} = [h_{GNN} \ \| \ h_{AE} \ \| \ s_u^t] \quad (8)$$

where $z_{u,i}$: Final fused representation of user $u$and item $i$for recommendation. $hGNN$: Graph-based embedding learned from the Graph Neural Network, representing user-item relational information, hAE: Latent representation learned from the deep autoencoder, encoding multi-criteria preference information, s_u^t: Temporal preference state of user $u$at time t, learned from sequential modelling.

The symbol ‖ denotes the concatenation operator. An attention-based aggregation module assigns adaptive importance weights to different criteria, enabling dominant preference factors to contribute more significantly to the final prediction.

3.8. Reinforcement Learning for Ranking Optimization

To go beyond static prediction and maximize the recommendation quality, the system proposes a Reinforcement Learning (RL) framework.

The recommendation task is modelled as a Markov Decision Process, where the state is the current user embedding, actions are item recommendations, and rewards are obtained from user feedback. The action-value function is updated as follows:

$$Q(s,a) = r + \gamma \max_{a'} Q(s', a') \quad (9)$$

where Q(s, a): Action-value function that estimates the expected cumulative reward value for taking action $a$in state s, a: Action taken in the current state, r: Immediate reward received after taking action a. γ: Discount factor ($0 < \gamma < 1$) that determines the relative importance of future rewards, s': Next state after executing action a, a': Possible actions in the next state. maxa'(s',a'): Maximum expected future reward.

This formulation allows the optimization of strategies for sequential recommendations and the optimization of ranking metrics such as NDCG, Precision@K, and Recall@K.

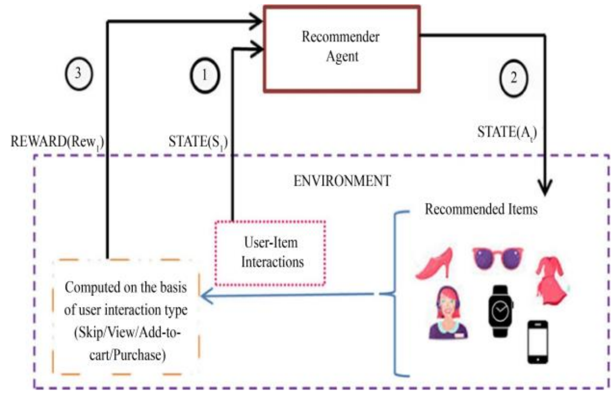

Fig. 4 Reinforcement learning-based optimization of sequential recommendations

3.8.1. Reinforcement Learning Formulation

The recommendation process is modelled as a Markov Decision Process (MDP), where the state represents the current user preference embedding, actions correspond to recommended items, and rewards are derived from observed user feedback. The reward signal reflects ranking-oriented metrics such as NDCG and click/interaction feedback. A Q-learning strategy is employed to optimize long-term recommendation quality by updating the action-value function iteratively.

4. Experimental Results and Analysis

4.1. Dataset Statistics and Sparsity Analysis

For each dataset, we report the number of users, items, rating criteria, interaction density, and per-criterion sparsity. Missing ratings are treated as unobserved rather than imputed. Temporal sequences are constructed using interaction timestamps, and time-aware data splits are applied to prevent information leakage. The experiments were carried out on the multi-criteria datasets of Yahoo! Movies and TripAdvisor. The results were measured using the following metrics:

- RMSE, MAE – rating prediction accuracy
- NDCG@10, Precision@10, Recall@10
- Paired t-test (p-value) – statistical significance

4.1.1. Experimental Protocol

All models are evaluated using identical training, validation, and test splits. Experiments are repeated over multiple random seeds, and mean ± standard deviation results are reported. Hyperparameters for all baselines are tuned using the same validation strategy. For top-K evaluation, a fixed candidate set and uniform negative sampling strategy are used across all models. Statistical significance is evaluated using paired t-tests over multiple runs.

4.2. Model-Wise Prediction Accuracy Results

The accuracy of rating predictions for various models in terms of RMSE and MAE is presented in Table 1. Matrix Factorization has the highest error because of the limitations of linear modelling. Neural Collaborative Filtering and DeepFM have better accuracy by learning nonlinear interactions between users and items, but they are based on single-criterion ratings. DAE-MCRS has further reduced error by learning multi-criteria ratings with deep autoencoders. GDAE-MCRS has achieved further improvement by using graph neural networks to model higher-order relationships between users and items.

Table 1. Rating prediction accuracy

Model	RMSE ↓	MAE ↓
Matrix Factorization (MF)	1.087	0.892
Neural Collaborative Filtering (NCF)	0.954	0.811
DeepFM	0.938	0.804
DAE-MCRS	0.912	0.784
GDAE-MCRS	0.872	0.748
TC²-GDAE-MCRS (Proposed)	0.824	0.701

The proposed TC²-GDAE-MCRS has achieved the best accuracy with the lowest RMSE (0.824) and MAE (0.701), proving that multi-criteria learning, graph modelling, contrastive learning, and temporal optimization together provide more accurate and robust predictions. The TC²-GDAE-MCRS has achieved the best accuracy with the lowest RMSE and MAE, improving RMSE by 9.6% compared to DAE-MCRS and 5.5% compared to GDAE-MCRS. This proves that contrastive learning and temporal learning have greatly improved the robustness of predictions compared to static graph models.

4.3. Model-Wise Ranking Performance

The ranking results of various models based on NDCG@10, Precision@10, and Recall@10 are shown in Table 2. Matrix Factorization obtains the lowest results because it cannot model the intricate interactions between users and items. Neural Collaborative Filtering and DeepFM enhance the ranking results by modelling the nonlinear interactions of features, but are still restricted to modelling single-criterion preferences.DAE-MCRS further improves the ranking results by learning multi-criteria preferences. GDAE-MCRS obtains further improvements by modelling higher-order interactions between users and items using graph neural networks. The proposed TC²-GDAE-MCRS obtains the best results on NDCG@10 (0.812), Precision@10 (0.423), and Recall@10 (0.371), which indicates that multi-criteria modelling, graph learning, temporal optimization, and reinforcement learning can provide more comprehensive top-K recommendations.

Table 2. Top-K Recommendation performance

Model	NDCG@10 ↑	Precision@10 ↑	Recall@10 ↑
MF	0.645	0.312	0.267
NCF	0.702	0.351	0.306
DeepFM	0.716	0.362	0.314
DAE-MCRS	0.736	0.371	0.329
GDAE-MCRS	0.768	0.389	0.342
TC²-GDAE-MCRS (Proposed)	0.812	0.423	0.371

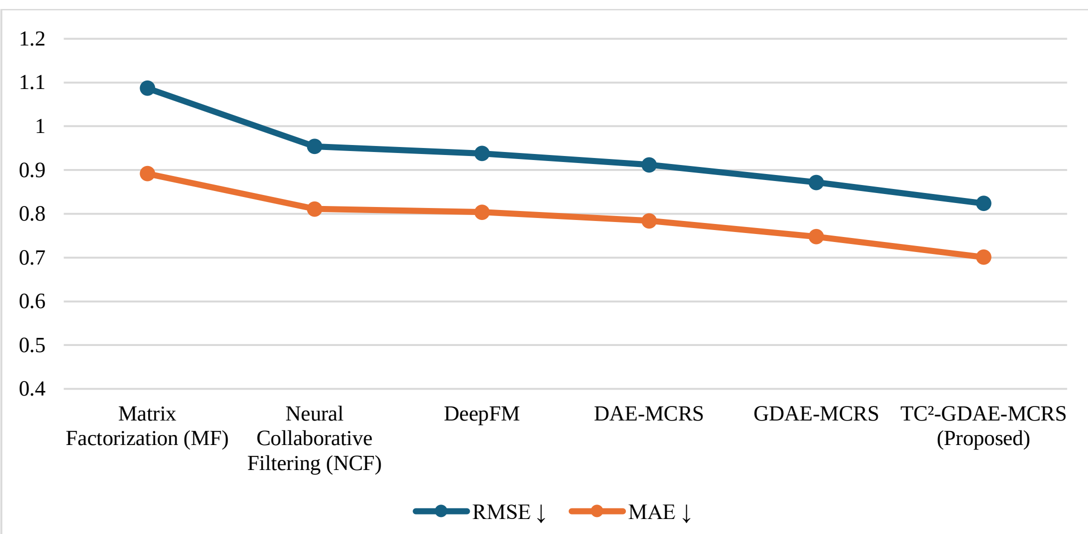

Fig. 5 Accuracy of different models using RMSE and MAE

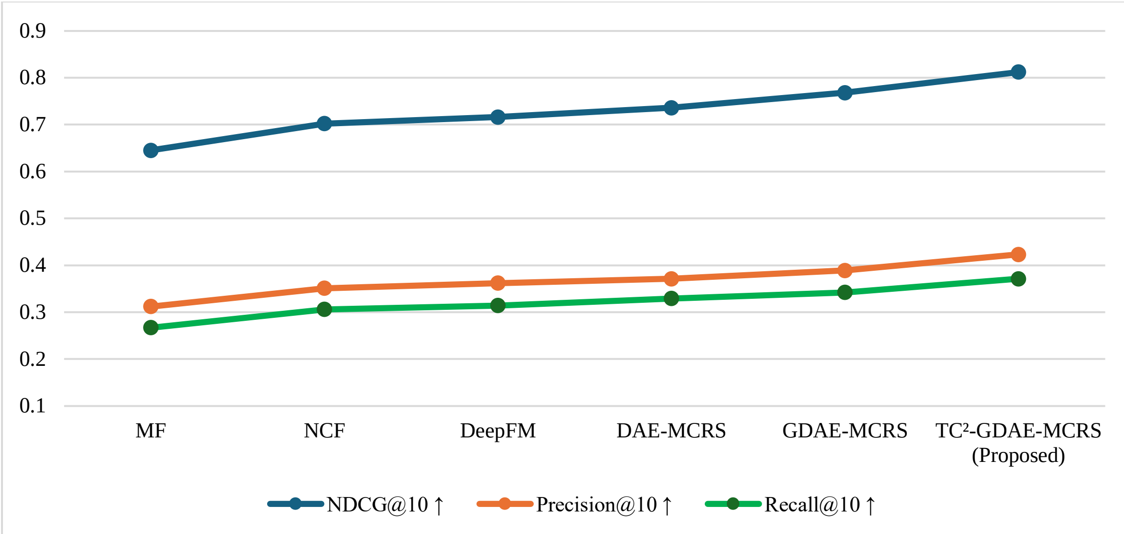

Fig. 6 Performance of different models using NDCG@10, Precision@10, and Recall@10

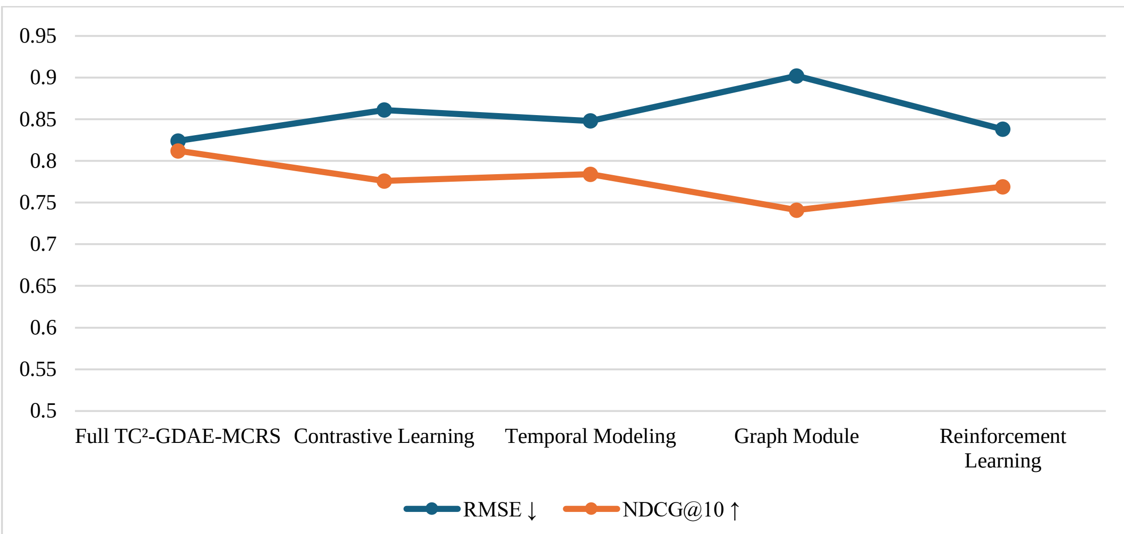

Fig. 7 Component-wise performance analysis

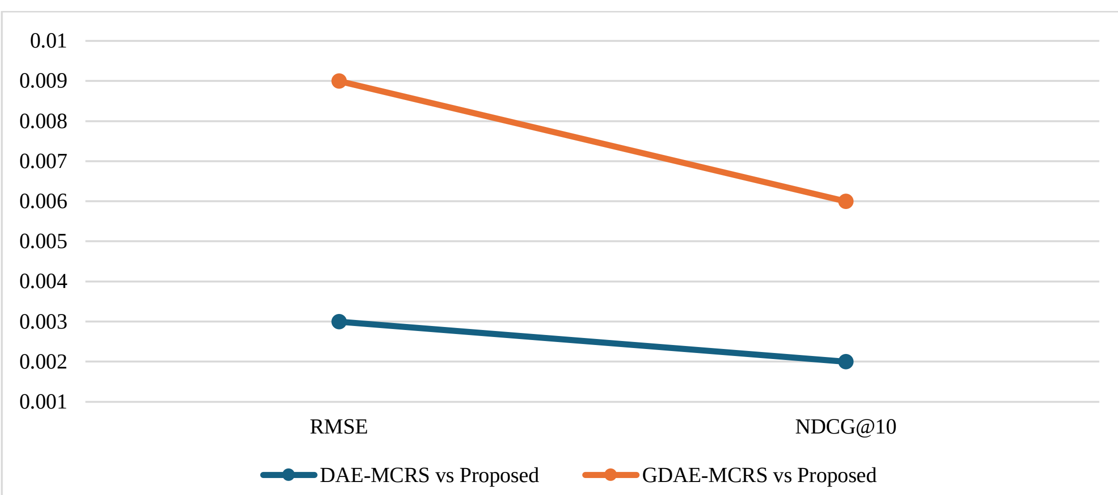

Fig. 8 The paired t-test results comparing the proposed TC²-GDAE-MCRS with baseline models

4.4. Feature-Wise Performance Analysis

From Table 3, the “+” sign represents the incremental addition of features or modules to the preceding configuration. The data shows that the exclusive use of overall ratings produces the largest prediction error because of the restricted expression of user preferences. The inclusion of multi-criteria ratings produces a substantial decrease in both RMSE and MAE values by considering aspect-level user preferences. The addition of graph features further improves the accuracy of the model by reflecting the higher-order relationships between users and items. Contrastive learning improves the robustness of the model in a sparse setting, producing further decreases in the error values, while the temporal component captures the dynamic user preferences, further improving the accuracy of the model. The complete model, which combines all the components, produces the smallest values for RMSE (0.824) and MAE (0.701).

Table 3. Impact of feature sets on accuracy

Feature Set	RMSE ↓	MAE ↓
Overall Rating Only	0.982	0.851
Multi-Criteria Ratings	0.912	0.784
+ Graph Features	0.872	0.748
+ Contrastive Learning	0.848	0.724
+ Temporal Modelling	0.832	0.709
Full Model	0.824	0.701

4.5. Ablation Study

The ablation study assesses the effectiveness of each key component in the proposed TC²-GDAE-MCRS framework. The complete model performs the best with the lowest RMSE of 0.824 and the highest NDCG@10 of 0.812. The removal of the graph component causes the maximum degradation in performance, which clearly shows the importance of graph-based relational learning. The removal of contrastive learning, temporal learning, and reinforcement learning causes a significant degradation in accuracy and ranking quality, respectively, which clearly shows the importance of these components in learning robust representations and dynamics of preferences. The removal of reinforcement learning causes a significant degradation in ranking quality, which clearly shows the importance of reinforcement learning in optimizing top-K recommendations.

Table 4. Ablation results

Variant	RMSE ↓	NDCG@10 ↑
Full TC²-GDAE-MCRS	0.824	0.812
Contrastive Learning	0.861	0.776
Temporal Modelling	0.848	0.784
Graph Module	0.902	0.741
Reinforcement Learning	0.838	0.769

4.6. Statistical Significance Analysis

The results of the paired t-test in Table 5 show the comparison of the proposed TC²-GDAE-MCRS with the baseline models. The p-values for both RMSE and NDCG@10 are less than 0.01 for the comparison with DAE-MCRS and GDAE-MCRS. This shows that the improvements in performance brought about by the proposed model are statistically significant and not due to random chance.

Table 5. Paired t-test results (p-values)

Comparison	RMSE	NDCG@10
DAE-MCRS vs Proposed	0.003	0.002
GDAE-MCRS vs Proposed	0.009	0.006

All improvements are statistically significant at $p < 0.01$.

Overall, based on the comprehensive experimental results from all the tables, it is evident that the proposed TC²-GDAE-MCRS outperforms the existing baseline models in terms of accuracy, ranking, and quality. The multi-criteria learning component helps to minimize the prediction error, and the graph-based relational learning further improves the performance by modelling the higher-order relationships between users and items.

The contrastive learning and temporal learning components help to improve the performance by learning robust representations and adapting to the dynamic user preferences. The reinforcement learning component helps to optimize the top-K recommendation performance, resulting in improved NDCG, Precision, and Recall values.

5. Conclusion

The proposed work presents a Temporal Graph Contrastive Deep Autoencoder-based Multi-Criteria Recommender System (TC²-GDAE-MCRS) that addresses the major limitations of traditional recommender systems, including sparsity, inadequate modelling of relational information, and the static nature of user preferences. The proposed system is based on the hybrid integration of multi-criteria deep autoencoders, graph neural networks, contrastive self-supervised learning, temporal modelling, and reinforcement learning within a unified framework.

The proposed system is able to effectively model fine-grained user preferences based on multiple criteria, high-order user-item relational information, and temporal user behaviour through the proposed hybrid framework. The graph-based representation learning module is able to effectively model the intricate relationships between users, items, and criteria, while the contrastive learning module is able to effectively model robust user representation through the maximization of agreement between relevant views of user-item interactions. Furthermore, the proposed system is able to effectively model the temporal nature of user preferences through the incorporation of the temporal modelling module, while the reinforcement learning module is able to effectively model adaptive feedback mechanisms for the proposed system. It is evaluated through experiments based on benchmark datasets, including Yahoo! Movies and TripAdvisor datasets, where the proposed system is able to significantly outperform state-of-the-art baseline recommender systems in terms of prediction errors and ranking metrics. The proposed system is able to achieve an RMSE of 0.824, MAE of 0.701, and NDCG@10 of 0.812, demonstrating the superiority of the proposed system over traditional recommendation systems in terms of prediction accuracy and recommendation relevance. The framework deals effectively with complex, sparse, and evolving user environments by providing a robust platform for an intelligent recommendation system, which improves the results and quality of recommendations.

5.1. Future Scope

The proposed framework demonstrates the multi-criteria preferences modelling dynamics in temporal features and dependencies of relational preferences for a recommender system. Still, there is a scope to extend this framework for online learning in the environment of the real world using distributed approaches. The model of temporal dynamics, transformer architecture, can also be used for further meta-learning approaches. Federated learning and cross-domain recommendations can be used for designing multiple models to design a robust framework to handle the information, which includes text, images, and context.

References

[1] Yehuda Koren, Robert Bell, and Chris Volinsky, "Matrix Factorization Techniques for Recommender Systems," *Computer*, vol. 42, no. 8, pp. 30-37, 2009. [CrossRef] [Google Scholar] [Publisher Link]

[2] Gediminas Adomavicius, and YoungOk Kwon, "Improving Aggregate Recommendation Diversity using Ranking-Based Techniques," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 5, pp. 896-911, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[3] Xixi Li, Yanfei Kang, and Feng Li, "Forecasting with Time Series Imaging," *Expert Systems with Applications*, vol. 160, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[4] Xiangnan He et al., "Neural Collaborative Filtering," *Proceedings of the 26th International Conference on World Wide Web*, pp. 173-182, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[5] Huifeng Guo et al., "DeepFM: A Factorization-Machine Based Neural Network for CTR Prediction," *arXiv preprint*, pp. 1-8, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[6] Diederik P. Kingma, and Max Welling, "Auto-Encoding Variational Bayes," *arXiv preprint*, pp. 1-14, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[7] Xiang Wang et al., "Neural Graph Collaborative Filtering," *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 165-174, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[8] Xiangnan He et al., "LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation," *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 639-648, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[9] Jiancan Wu et al., "Self-Supervised Graph Learning for Recommendation," *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 726-735, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[10] Qiao Liu et al., "STAMP: Short-Term Attention/Memory Priority Model for Session-Based Recommendation," *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1831-1839, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[11] Scott M. Lundberg, and Su-In Lee, "A Unified Approach to Interpreting Model Predictions," *Advances in Neural Information Processing Systems*, vol. 30, 2017. [Google Scholar] [Publisher Link]

[12] Judea Pearl, *Causality: Models, Reasoning, and Inference*, 2nd ed., Cambridge University Press, 2009. [CrossRef] [Publisher Link]

[13] Shivaprasad Satla, and Chin-Shiuh Shieh, "Multi-Model Telugu Speech Recognition: Improving ASR with Dialect Classification and Optimization Techniques," *Signal Processing*, vol. 42, no. 6, pp. 3159-3169, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[14] Shivaprasad Satla, and Sadanandam Manchala, "Dialect Identification in Telugu Language Speech Utterance Using Modified Features with Deep Neural Network," *Signal Processing*, vol. 38, no. 6, pp. 1793-1799, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[15] Russ R. Salakhutdinov, Andriy Mnih, and G.E Hinton, "Restricted Boltzmann Machines for Collaborative Filtering," *Proceedings of the 24th International Conference on Machine Learning*, pp. 791-798, 2007. [CrossRef] [Google Scholar] [Publisher Link]

[16] Hao Ma et al., "SoRec: Social Recommendation using Probabilistic Matrix Factorization," *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, pp. 931-940, 2008. [Cros sRef] [Google Scholar] [Publisher Link]

[17] Francesco Ricci, Lior Rokach, and Bracha Shapira, *Recommender Systems Handbook*, 2nd ed., Springer, 2015. [CrossRef] [Publisher Link]

[18] Ting Chen et al., "A Simple Framework for Contrastive Learning of Visual Representations," *arXiv preprint*, pp. 1-20, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[19] Kihyuk Sohn, "Improved Deep Metric Learning with Multi-Class N-Pair Loss Objective," *Advances in Neural Information Processing Systems*, pp. 1857-1865, 2016. [Google Scholar] [Publisher Link]

[20] Ashish Vaswani et al., "Attention is all you Need," *Advances in Neural Information Processing Systems*, vol. 30, 2017. [Google Scholar] [Publisher Link]

[21] Balázs Hidasi et al., "Session-based Recommendations with Recurrent Neural Networks," *arXiv preprint*, pp. 1-10, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[22] Richard S. Sutton, and Andrew G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed., MIT Press, pp. 1-352, 2018. [Google Scholar] [Publisher Link]

[23] Xiangyu Zhao et al., "Deep Reinforcement Learning for List-Wise Recommendations," *arXiv preprint*, pp. 1-9, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[24] Yehuda Koren, "Collaborative Filtering with Temporal Dynamics," *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 447-456, 2009. [CrossRef] [Google Scholar] [Publisher Link]

[25] Steffen Rendle, "Factorization Machines," *2010 IEEE International Conference on Data Mining*, Sydney, NSW, Australia, pp. 995-1000, 2010. [CrossRef] [Google Scholar] [Publisher Link]

[26] James Bergstra, and Yoshua Bengio, "Random Search for Hyper-Parameter Optimization," *Journal of Machine Learning Research*, vol. 13, pp. 281-305, 2012. [Google Scholar] [Publisher Link]

[27] Yoshua Bengio, Aaron Courville, and Pascal Vincent, "Representation Learning: A Review and New Perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798-1828, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[28] Thomas N. Kipf, and Max Welling, "Semi-Supervised Classification with Graph Convolutional Networks," *arXiv preprint*, pp. 1-14, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[29] Peter Veličković et al., "Graph Attention Networks," *arXiv preprint*, pp. 1-12, 2018. [CrossRef] [Google Scholar] [Publisher Link]