Context-Enriched Temporal Knowledge Graph with Reinforcement Learning for Adaptive Music Recommendation in Entertainment Platforms

Pushpa Mohan[1*], *Leo* Gertrude David[2], *Ayesha* Siddiqua[3], and *Ashok* Kumar Pamidi Venkata[4]

[1]Department of Computer Science and Engineering, HKBK College of Engineering, Bengaluru, India
[2]Department of Visual Communication, Kumaraguru College of Liberal Arts and science, Coimbatore, India
[3]Department of Electronics and Communication Engineering, Nitte Meenakshi Institute of Technology, Nitte (Deemed to be University), Bengaluru
[4]Megan Soft INC, Livonia, United States

Abstract. Currently, Music Recommendation (MR) platforms are involved in exploiting meaning-based representations of music data to deliver personalized recommendations. These models enhance music discovery by analyzing user preferences, song attributes, and contextual data. However, the existing frameworks heavily depend on static user-item interactions and knowledge graphs, which leads to overlooking dynamic factors that reduce adaptivity in real-world entertainment platforms. Hence, this study proposes a Context-enriched Temporal Knowledge Graph with Reinforcement Learning (CTKG-RL) for adaptive MR in entertainment platforms. Initially, the input data were collected from the Last.FM_ 1 K dataset, which contains user-track interactions and then preprocessed with timestamp conversion, Levenshtein string matching, and thresholding-based filtering. This process ensures that the model has consistent time alignment and deduplication of track metadata and retains only active users and items, respectively. Further, a TKG is constructed from semantic and contextual triples, and then the embeddings are learned using Temporal Graph Attention Networks (TGANs) to capture evolving user interests. Subsequently, session modeling with a Graph Recurrent Unit (GRU)-based encoder aggregates short- and long-term preferences, and finally, an actor-critic RL model optimizes adaptive playlist generation. The proposed CTKG-RL achieved better results in terms of accuracy (0.893) when compared to the existing Multi-channel and Multi-loss MR Knowledge Graph (MM-MRKG) model.

1 Introduction

In recent years, music streaming platforms provides users with their access to millions of songs users has often faces difficulties in finding personally preferred music due to the overwhelming volume of available content [1]. Moreover, the traditional music recommendation systems rely filtering methods which offers useful recommendations but struggles with some problems. Furthermore, the collaborative filtering compares user behaviors and listening histories to suggest new tracks but it fails when users or songs have limited historical data [2]. Eventually, popularity bias has caused systems to repeatedly recommend only mainstream songs by reducing diversity [3]. Also, to overcome these challenges researchers has increasingly adopted Knowledge Graphs (KGs). Where, these graphs represent music related entities such as songs, albums, artists, genres, moods, labels and their relationships in a structured way by enabling richer contextual information for recommendations [4] Moreover, strong popularity bias existed where the recommendation of the algorithms repeatedly suggested the most popular or mainstream tracks by neglecting niche or less known artists and also reduced diversity in recommendation. Also, ensured diversity and novelty remained a challenge since the users often wished to discover new artists, genres, or music styles, but algorithms inclined to suggest items [5]. Furthermore, explainability of recommendations are limited as many models was operated as black boxes and failed to justify why a particular song or artist was recommended that reduced user trust and acceptance. Therefore, dynamic user preferences were difficult to handle as a user's taste on music changed over time depending on mood, trends, or context, yet many systems struggled to adapt to these temporal shifts [6].

However, the cold start problem is addressed by incorporating auxiliary information like textual descriptions, social data, and knowledge graph entities by linking new songs and users in the KG where the model has the ability to generate recommendations even with minimal interaction data [7]. Moreover, the issue of data sparsity is reduced by leveraging knowledge graphs that provide richer semantic relationships among music items, such as artist-genre album links. In addition, this additional context helps the algorithms to fill the gaps left by sparse user–item interaction

* Corresponding author: acharyamohan036@gmail.com

matrices. Furthermore, the problem of popularity bias is mitigated by designing algorithms that explicitly consider novelty and diversity, such as bounded greedy selection strategies and diversity-enhanced ranking methods, to ensure that recommendations include both popular and lesser-known items [8]. Eventually, to overcome the lack of diversity and novelty in systems, collaborative knowledge graphs with graph embeddings are combined. This combination slows users to discover styles, new bands, and genres by exploring semantic similarities rather than relying solely on past behavior [9]. Hence, the challenge of explainability is solved by using knowledge graphs enriched with semantic metadata, where these graphs enable the systems to provide clear reasoning; for example, this artist is recommended because they share a genre or label with an artist you like, which increases users' trust [10]. The state-of-the-art method follows knowledge Graph Embedding-Based Recommendation (EARS), which is designed to transform music-based entities. Moreover, from a knowledge graph to vector embeddings, it captures the semantic relationships among entities and enables similarity calculations between users and items in a latent space [11]. Eventually, the embeddings were used to recommend new artists or songs by finding items that are closest in the vector space to the user's listening history. Hence, the quality of recommendations heavily depended on the completeness and accuracy of the knowledge graph where the graph was sparse, outdated, and contained noisy links where the embeddings will generate misleading recommendation [12].

The main contributions of this research are as follows:

- The proposed Context-enriched Temporal Knowledge Graph with Reinforcement Learning (CTKG-RL) framework delivers adaptive music recommendations by embedding contextual and temporal signals into graph for ensuring the dynamic personalization.
- A Gated Recurrent Unit (GRU)-based encoder is employed as a session modeling technique to fuse short-term listening behaviors with long-term preference patterns to provide playlist recommendations that reflect immediate user intent to improve user satisfaction.
- The actor-critic Reinforcement Learning (RL) strategy optimizes playlist generation through reward-driven adaptation by balancing the diversity and satisfaction objectives to provide recommendations that are accurate and improve music discovery.

The structure of this paper is organized as follows: existing works and their drawbacks are presented in Section 2; the overall explanation of the methodology is described in Section 3; the experimental results and discussion are presented in Section 4; and Section 5 concludes the paper.

2 Literature review

Existing studies on semantic knowledge modeling for adaptive music recommendation in entertainment platforms are as follows. Xinqiao Liu et al. [13] introduced Multi-channel and Multi-loss Music Recommendation Knowledge Graph (MM-MRKG) model based on knowledge graph and multi-task feature learning. Firstly, Last.FM, MovieLens-1M and Book-Crossing datasets were collected to capture user item interactions and enrich them with knowledge graph triples for more accurate recommendations. After that, the data was preprocessed which included data cleaning, knowledge graph construction and entity extraction techniques. Next, feature extraction was done where the features like user preferences, item attributes along with semantic relations were extracted to create meaningful low dimensional vectors which captured hidden patterns. Finally, the model's integrated the knowledge graphs with recommendation modules which enhanced personalization, accuracy, and robustness by addressing cold-start and data sparsity issues. However, the model increased its complexity, dependence on the completeness and quality of the knowledge graph, and higher computational costs. Niels Bertram et al. [14] introduced I am all EARS using open data and knowledge graph embeddings for music recommendations by collecting MusicBrainz and Last.fm datasets to build a collaborative knowledge graph. After that, the data was preprocessed which involved data sampling to reduce dataset size and integration of both sources which was required to handle the large scale of data. Next, features extraction was done where the features of artist attributes such as genre, label, and origin along with user artist interaction data were extracted. Finally, the model has an advantage where enhanced recommendation quality, diversity, and explainability. However, the model faced few issues related to computational complexity and dependency on parameter tuning.

Aleksandar Ivanovski et al. [15] introduced knowledge graph-based recommender for automatic playlist continuation by collecting music recommendation dataset to build a personalized recommendation system. Next, the data was preprocessed by normalization, data cleaning techniques that ensured data consistency and enhanced model accuracy. After that, the features related to audio such as pitch, tempo, metadata and rhythm were extracted to enhance recommendation relevance and match similarity. Finally, the model captured both explicit and implicit user preferences which improved personalization, and handled large scale data efficiently. Hence, the model faced problems related to cold start where the new users or songs, potential bias in recommendations, and dependency on high quality feature extraction. Matej Bevec et al. [16] introduced Hybrid music recommendation with graph neural networks by collecting spotify graph dataset which combined the playlists and songs to capture relationships for music recommendation. After that, the data was preprocessed by randomizing search offsets technique to reduce popularity bias. Next, feature extraction was done where the features from metadata such as artist and popularity, along with deep audio embeddings were extracted where these features were used to represent songs in the graph by supporting

similarity learning. Finally, the model merged interaction data with content features which enables better handling of cold start problems. However, the model faced issues such as persistence of popularity error and high computational cost of graph models. Keigo Sakurai et al. [17] introduced controllable music playlist generation based on knowledge graph and reinforcement learning. Firstly, Spotify million playlist dataset was collected to predict preferences and generate playlists based on users' choice. After that, the data was preprocessed by splitting the playlists into training and testing sets also normalized popularity and year ranges. Next, features extraction was done which included extraction of acoustic features such as danceability, energy, tempo, and valence, along with popularity and novelty indices. Finally, the model predicted both long term and short-term user interests and allowed controllable playlist generation. However, the model faced issues related to dependency on metadata quality that lack the real user's evaluation also limited reward function factors.

3 Methodology

This study proposes a CTKG-RL framework for adaptive music recommendation in entertainment platforms. This process begins with data collection from the Last.FM_ 1 K dataset, which contains user-track interactions. The collected data are then preprocessed with timestamp conversion, Levenshtein string matching, and thresholding-based filtering to ensure consistent time alignment, duplicate track metadata, and retain active users as well as items. Furthermore, a TKG is constructed from semantic and contextual triples, and embeddings are learned using Temporal Graph Attention Networks (TGANs) to capture evolving user interests. Subsequently, session modeling was performed using a GRU-based encoder that aggregates short- and long-term preferences. Finally, an actor-critic RL model is employed to optimize adaptive playlist generation to allow the system to adapt to user preferences and provide personalized music recommendations. An explanation of the proposed CTKG-RL is shown in Figure 1.

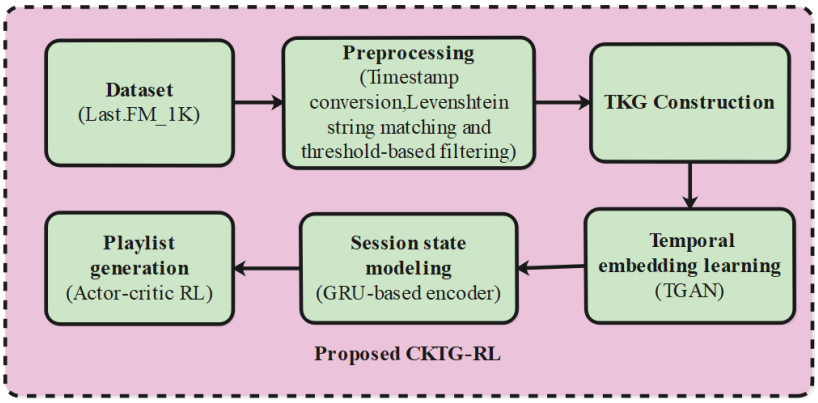

Fig. 1. Pictorial representation of proposed CKTG-RL.

3.1 Dataset

The proposed CTKG-RL framework involves in data collection from Last.FM_1K dataset [18] which contains listening histories of one thousand users over an extended period. Each record consists of a user identifier, the track played and a timestamp indicating the exact listening event. Such information captures both the preferences of individual listeners and the temporal sequence of their music consumption patterns. From these raw logs, an interaction structure is derived in which users are represented as rows, tracks as columns and listening events as temporal markers. Here, the dataset is mathematically expressed as a three-dimensional relation which is given in Equation (1):

$$Y = y_{u,i,\tau} \quad (1)$$

Where, $y_{u,i,\tau} = 1$ indicates that user u has played track i at time τ respectively. This formulation preserves the temporal dimension for enabling subsequent modeling approaches to explore how users' interests evolve over time. The collected data is further fed into preprocessing which is clearly explained in next section.

3.2 Preprocessing

Further, the organized interaction records are given as input to the preprocessing which is performed in three stages for ensuring reliability of modeling. Initially, each listening event's time field is converted into a uniform epoch format which guarantees precise temporal alignment across users and sessions. Then, the Levenshtein string matching is employed to handle redundancy in track and artist names for enabling the detection and merging of near-duplicate entries while preserving the semantic integrity of the music catalog. Finally, the thresholding-based filtering is employed to eliminate users with insufficient listening histories and items with extremely low frequency. The operation of these techniques is described in the following subsections.

3.2.1 Timestamp conversion

Initially, preprocessing involves normalization, where the listening events in the dataset are logged in heterogeneous formats; thus, aligning them into a

uniform representation is crucial to preserve chronological integrity across users. Here, the UNIX epoch time conversion technique was used, where each timestamp was mapped to the number of seconds elapsed. Generally, for any recorded listening event with calendar representation (d, m, y, h, min, s), the conversion function is expressed as Equation (2):

$$\tau = (y - 1970) \times T_y + m \times T_m + d \times T_d + h \times 3600 + min \times + s \quad (2)$$

Where, T_y, T_m and T_d define the equivalent number of seconds for one year, month, and day, respectively. This transformation ensures that every play event is captured as a precise scalar value that reflects its position on a continuous time-axis. The main advantage of epoch conversion is that it discards ambiguities due to different regional formats, daylight adjustments, and time-zone inconsistencies for temporal modeling. The final output is a dataset where all listening events are represented as consistent epoch values by providing accurate chronological order.

3.2.2 Levenshtein string matching

Further, the Levenshtein distance-based string matching is employed to handle the inconsistencies in track and artist naming by quantifying the dissimilarity between two strings by calculating minimum number of single-character operations required to transform one into the other. These operations include insertion, deletion and substitution. The Levenshtein distance between two strings a and b of lengths $|a|$ and $|b|$ is formally defined as Equation (3):

$$D(i,j) = min \begin{cases} D(i-1,j) + 1, \\ D(i,j-1) + 1, \\ D(i-1,j-) + [a_i \neq b_j] \end{cases} \quad (3)$$

Where, $D(0,j) = j, D(i,0) = i$ and $[a_i \neq b_j]$ equals 0 if the characters match and 1 otherwise respectively. The final output is a refined catalog of tracks and artists, in which duplicates and inconsistent labels are consolidated into unique identifiers to ensure semantic clarity.

3.2.3 Threshold-based filtering

In the dataset, the user interaction logs contain long tails of extremely sparse activity, where some users listen to only a handful of tracks, and certain items appear only a few times. Retaining these data introduces noise and weakens the generalization ability of the embedding models. Hence, threshold-based filtering is employed, which systematically removes users and items that do not meet the minimum frequency requirement. Let I_u defines the total number of interactions for user u and U_i denote the total number of interactions for item i respectively. The filtering rule is expressed by Equation (4):

$$U' = \{u | I_u \geq \theta_u\}, \quad I' = \{i | U_i \geq \theta_i\}, \quad (4)$$

Where θ_u and θ_i are the predefined thresholds for user activity and item popularity, respectively. Only the users and items that satisfy these constraints are retained by producing a denser interaction matrix. For example, setting $\theta_u = 20$ and $\theta_i = 50$ ensures that each user has a sufficient history to model preferences, where each item maintains sufficient exposure to learn a reliable representation. This selective process removes weak signals and balances the distribution of interactions. The final output is concatenated interaction data that preserve only active users and frequently consumed items.

3.3 TKG construction

Furthermore, the proposed CTKG-RL framework involves building a structured representation that captures the semantic relations and contextual dynamics of user interactions. Here, the Ontology-Guided TKG construction technique, which integrates music metadata with contextual attributes into a graph-based schema, is employed. Each listening record is then expressed as a quadruple (h, r, t, τ) where h and t defines the head and tail entities, r signifies the semantic relation, and τ encodes the timestamp. For example, a record is mapped as $(User, listens\_to, Track, \tau)$ while the additional metadata extends the structure with edges such as $(Track, hasGenre, Genre, \tau)$. The inclusion of the temporal variable τ transforms a static graph into a temporal knowledge graph to enable the system to model the structural relations and their evolution over time. Formally, TKG is denoted by Equation (5):

$$\mathcal{G}_\tau = \{(h, r, t, \tau) | h, t \epsilon \varepsilon, r \epsilon R, \tau \epsilon \mathbb{T}\} \quad (5)$$

Where ε represents the entities, R denotes the relations, and $\mathbb{T}$ is the set of discrete time points. The final output is a constructed TKG graph that encodes users, items, and contextual attributes as temporally grounded semantic triplets.

3.4 Temporal embedding learning with TGAN

Once the TKG is constructed, the next step is to learn meaningful vector representations that preserve both semantic relations and their evolution across time. Here, TGAN is employed, which extends graph attention mechanisms by incorporating temporal encodings into the aggregation process. For a given quadruple (h, r, t, τ), the embedding of entity h at time τ is updated by attending to its neighbors $\mathcal{N}(h)$ with temporal weights. Then, the update is formulated as shown in Equation (6):

$$e_h^\tau = \sigma\left(\sum_{(h,r_i,t_i,\tau_i)\epsilon\mathcal{N}(h)} \alpha_{h,t_i}^{\tau,\tau_i} . W . \left[e_{t_i}^{\tau_i} || r_i || \phi(\Delta\tau)\right]\right) \quad (6)$$

Where W is a transformation matrix, r_i defines relation embedding, $\phi(\Delta\tau)$ encodes the time difference between τ and τ_i, and $\alpha_{h,t_i}^{\tau,\tau_i}$ are the learned attention

coefficients capturing the relevance of neighbor t_i respectively. This design allows the model to highlight more recent or contextually aligned interactions while still retaining historical influences. Then, the network generates embeddings that dynamically adapt to shifts in user preferences and item popularity patterns by iteratively applying these updates. These embeddings form the semantic-temporal backbone required for capturing adaptive behaviors in recommendation. The final output is a set of time-aware embeddings for user, tracks and contextual entities by providing dynamic feature representations.

3.5 Session state modeling with GRU-based encoder

Furthermore, the GRU encoder is employed to capture the evolving intent of listeners during a session to model the sequential dependencies while maintaining efficiency. Here, a user's recent listening sequence is treated as an ordered set of temporal embeddings, $\left\{e_{t_1}^{\tau_1}, e_{t_2}^{\tau_2}, \ldots, e_{t_k}^{\tau_k}\right\}$. At each step, the GRU updates its hidden state using recurrence, which is expressed in Equation (7):

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{7}$$

Where the update gate $z_t = \sigma\left(W_z e_{i_t} + U_z h_{t-1}\right)$ controls how much of the previous state is retained and the candidate hidden state integrates the current input with the past context, which is defined in Equation (8):

$$\tilde{h}_t = \tanh\left(W_h e_{i_t} + U_h(r_t \odot h_{t-1})\right) \tag{8}$$

This mechanism balances long-term preference memory with short-term shifts by producing a compact session representation vector s_τ. Then, the GRU encoder effectively captures patterns such as mood-driven bursts, genre shifts and repeated plays by aggregating multiple interactions. The final output is a session state vector s_τ which summarizes both short-term dynamics and long-term preferences.

3.6 Playlist generation with actor-critic RL model

Finally, the proposed CTKG-RL framework transforms session representations into adaptive recommendations by framing playlist generation as a sequential decision-making problem. Here, the actor-critic RL model is employed to balance policy exploration with value estimation. At each interaction, the session state vector s_τ derived from the GRU encoder serves as the environment state, whereas the recommendation of a track constitutes the action. The actor network $\pi_\theta(a|s_\tau)$ outputs a probability distribution over candidate songs by guiding playlist formation, whereas the critic network estimates the value function $V^\pi(s_\tau)$ by representing the expected long-term reward from the state. The optimization objective is expressed in Equation (9):

$$J(\theta) = \mathbb{E}_{\pi_\theta}[\sum_{t=0}^{T} \gamma^t r_t] \tag{9}$$

Where r_t denotes the reward at step t by capturing engagement signals, such as repeated plays, diversity of selections, and overall user satisfaction, and γ represents the discount factor. The actor then improves its policy, while the critic refines value estimates through iterative updates, resulting in a stable learning loop that adapts to dynamic user contexts. The final output is an adaptive playlist generation policy π_θ which continuously adjusts user behavior by producing context-aware and satisfaction-driven music recommendations.

4 Experimental results

In this study, an adaptive music recommendation on entertainment platforms with the help of CTKG-RL was presented by implementing the Python package PyTorch. Software and hardware requirements were used for the experimental analysis. The software requirements include the Windows 11 Pro operating system and hardware requirements such as Intel Core i4-8100 processor with 16.00 GB RAM and CPU 3.63GHz processing speed. To evaluate the proposed CTKG-RL, Accuracy (ACC), Precision (PR), Recall (RE), and Area Under Curve (AUC) were used as the evaluation metrics. which are expressed in Equations (10)-(13) as follows:

$$Accuracy\ (Acc) = \frac{A+B}{A+B+C+D} \tag{10}$$

$$Precision\ (PR) = \frac{A}{A+C} \tag{11}$$

$$Recall\ (RE) = \frac{A}{A+D} \tag{12}$$

$$AUC = \int_a^b f(x)dx \tag{13}$$

Here, A is true positive, B represents true negative, C defines false positive as well as D signifies false negative respectively. Similarly, a, b are the limits and $f(x)$ is a function.

4.1 Performance analysis

The performance analysis of the proposed CTKG-RL for adaptive music recommendation in entertainment platforms was performed by using various traditional models such as the Multi-task Feature Learning for Graph Enhanced Recommendation (MFL-GER), Collaborative Knowledge Base Embedding for Recommender Systems (CKBE-RS) and Ripple Network for Knowledge Graph Reasoning in RS (RippleNet) with the help of error metrics. A graphical representation of the performance of the proposed CTKG-RL is shown in Figure 2.

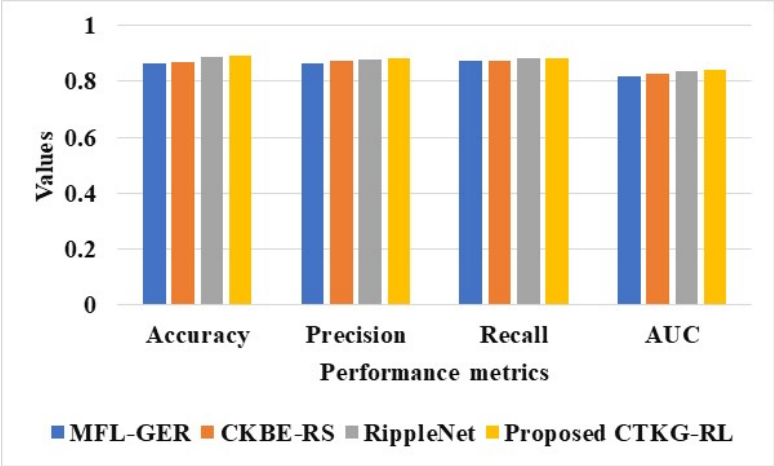

Fig. 2. Performance analysis of proposed CTKG-RL.

Figure 2 shows the performance analysis of the proposed CTKG-RL with various traditional models including MFL-GER, CKBE-RS, and RippleNet. The proposed CTKG-RL provided better results with respect to the ACC, PR, REC, and AUC of 0.893, 0.885, 0.882, and 0.842, respectively.

4.2 Comparative analysis

The proposed CTKG-RL was estimated by comparing it with the existing MM-MRKG model [13]. A comparison of the proposed CTKG-RL was performed with the help of the evaluation metrics ACC and AUC, which are explained in Table 1.

Table 1. Comparative analysis of proposed CTKG-RL.

Comparative models	Accuracy	Area Under Curve
MM-MRKG [13]	0.763	0.816
Proposed CTKG-RL	0.893	0.842

Table 1 presents a comparative evaluation of the proposed CTKG-RL with the existing MM-MRKG [13] model using evaluation metrics. MM-MRKG [13] provided the best results with respect to ACC (0.763) and AUC (0.816). The results show that the proposed CTKG-RL outperformed the existing models, improving adaptive music recommendation in entertainment platforms. The proposed CTKG-RL achieved better results for ACC and AUC of 0.893 and 0.842, respectively, compared to existing models.

4.3 Discussion

The main objective of this study is to develop a system that exploits knowledge graphs and RL to provide personalized music recommendations that capture users' long-term preferences, adapt to their unique tastes, and improve music discovery. The results show that the proposed CTKG-RL effectively overcomes the drawbacks of the existing MM-MRKG [13] and traditional MFL-GER, CKBE-RS, and RippleNet by incorporating contextual information. Here, the existing MM-MRKG [13] heavily depended on multi-channel and multi-loss learning, which still treated user-item relations in a relatively static manner, limiting its ability to respond to rapid shifts in listening behavior. Thus, the proposed CTKG-RL overcomes this problem by embedding temporal dynamics directly into the knowledge graph and training with reinforcement signals to enable it to capture evolving interests in real time. Similarly, the traditional MFL-GER emphasizes multi-feature learning from graph embeddings but lacks a mechanism for adaptivity, where its recommendations remain fixed to historical associations. The proposed CTKG-RL learns both short-term bursts of preference and long-term stability by integrating GRU-based session modeling to personalize playlists more effectively. In addition, the CKBE-RS framework suffers from cold-start issues owing to sparse interactions and rigid knowledge base embeddings. This study addresses this issue by fusing contextual attributes into a temporal knowledge graph to provide richer semantic links that reduce sparsity effects. Moreover, RippleNet was strong in propagating user interests through layered ripples in the knowledge graph, although it amplified noise when irrelevant relations spread too far. Hence, the proposed CTKG-RL explores attention-based temporal embedding to assign higher weights to relevant recent interactions by minimizing the propagation of weak signals. Finally, the proposed CTKG-RL improves music discovery and personalization beyond what these earlier models achieved with a combination of temporal representation, contextual integration, and reinforcement-driven adaptability.

5 Conclusion

In this study, an efficient CTKG-RL framework was proposed for adaptive music-recommendation systems on entertainment platforms. Most existing models have the limitation of static user-item interactions; thus, this research overcomes this by incorporating dynamic factors and temporal information to enable adaptive music recommendations. In addition, the proposed CTKG-RL overcomes inconsistent data quality by using timestamp conversion, Levenshtein string matching, and thresholding-based filtering, which ensures

consistent time alignment. The construction of a TKF from semantic and contextual triples addresses static knowledge representation by capturing evolving user interests and preferences. Moreover, the use of TGANs overcomes the limited representation of dynamic user behavior by learning embeddings that capture temporal patterns. In addition, session modeling with a GRU-based encoder overcomes the insufficient aggregation of short- and long-term preferences for personalized music recommendations. Furthermore, an actor-critic RL model optimizes adaptive playlist generation, which helps overcome suboptimal recommendation strategies. From the results, the proposed CTKG-RL established better results in terms of accuracy (0.893) than existing models. In the future, this research will focus on investigating alternative RL techniques and evaluating the performance of the framework on larger datasets to improve its effectiveness and generalizability.

References

1. R. Behrens, A.K. Kupfer, T. Hennig-Thurau, There is business like show business! What marketing scholars and managers can learn from 40 years of entertainment science research. J. Acad. Mark. Sci. **53**, 760-780 (2025). https://doi.org/10.1007/s11747-024-01057-2
2. R. Garapati, M. Chakraborty, Recommender systems in the digital age: a comprehensive review of methods, challenges, and applications. Knowl. Inf. Syst., 1-45 (2025). https://doi.org/10.1007/s10115-025-02453-y
3. M. Pasupuleti, S.M. Satapathy, Unleashing the power of untuned large language models in recommender systems: a thorough investigation of current approaches, challenges, and future research directions. Knowl. Inf. Syst, 1-77 (2025). https://doi.org/10.1007/s10115-025-02539-7
4. A. Smajić, R. Karlović, M. Bobanović Dasko, I. Lorencin, Large language models for structured and semi-structured data, recommender systems and knowledge base engineering: a survey of recent techniques and architectures. Electron. **14**, 3153 (2025). https://doi.org/10.3390/electronics14153153
5. Q. Bsoul, F. Zawaideh, B.S. Alqadi, L.A. Almusfar, O.I. Khalaf, A.S. Alattas, M. Alali, D.S. AbdElminaam, From user preferences to accurate predictions: enhancing movie recommendation systems with neural collaborative filtering and sentiment analysis. SN Comput. Sci. **6**, 257 (2025). https://doi.org/10.1007/s42979-025-03742-7
6. K. Jain, R. Jindal, Optimization-based noise filtering among user-centric tweets to improve predictions in recommendation system. J. Supercomput. **81**, 1005 (2025). https://doi.org/10.1007/s11227-025-07457-7
7. C. Jin, M. Lin, F. Wu, X. Wu, Y. Zhou, J. Wang, TVMTrailer: a text-video-music AIGC framework for film trailer generation. IEEE Trans. Syst. Man Cybern. Syst. **55**, 6000-6010 (2025). https://doi.org/10.1109/TSMC.2025.3576988
8. T. Sachdeva, L.M. Goyal, M. Mittal, Mapping the landscape of personalization: a comprehensive review of prediction and trends in recommendation systems. Wiley Interdiscip. Rev. Data Min. Knowl. Discov. **15**, e70006 (2025). https://doi.org/10.1002/widm.70006
9. K. Sinha, D.S. Jana, Chapter 9 Multimedia database: data structures and algorithms for efficient management, 153-174 (2025).
10. S. Patil, R. Patil, S. Goudar, S. Sangani, R.H. Goudar, Review on music emotion analysis using machine learning: technologies, methods, datasets, and challenges. Discover Appl. Sci. **7**, 692 (2025). https://doi.org/10.1007/s42452-025-07178-9
11. A.D. Lokmanoglu, D. Walter, Topic modeling of video and image data: a visual semantic unsupervised approach. Commun. Methods Meas., 232-279 (2025). https://doi.org/10.1080/19312458.2025.2549707
12. L. Zhao, J. Yu, A.A. Laghari, K. Fang, T.R. Gadekallu, Feature jointly-based knowledge enhancement model for multimodal sentiment analysis. IEEE Trans. Comput. Soc. Syst. 1-13 (2025). https://doi.org/10.1109/TCSS.2025.3582541
13. X. Liu, Z. Yang, J. Cheng, Music recommendation algorithms based on knowledge graph and multi-task feature learning. Sci. Rep. **14**, 2055 (2024). https://doi.org/10.1038/s41598-024-52463-z
14. N. Bertram, J. Dunkel, R. Hermoso, I am all EARS: using open data and knowledge graph embeddings for music recommendations. Expert Syst. Appl. **229**, 120347 (2023). https://doi.org/10.1016/j.eswa.2023.120347
15. A. Ivanovski, M. Jovanovik, R. Stojanov, D. Trajanov, Knowledge graph based recommender for automatic playlist continuation. Information **14**, 510 (2023). https://doi.org/10.3390/info14090510
16. M. Bevec, M. Tkalčič, M. Pesek, Hybrid music recommendation with graph neural networks. User Model. User-Adap. Interact. 34, 1891-1928 (2024). https://doi.org/10.1007/s11257-024-09410-4
17. K. Sakurai, R. Togo, T. Ogawa, M. Haseyama, Controllable music playlist generation based on knowledge graph and reinforcement learning. Sens. **22**, 3722 (2022). https://doi.org/10.3390/s22103722
18. Last.FM_ 1 K dataset: https://www.kaggle.com/datasets/harshal19t/lastfm-dataset (Accessed on July 2025)