Time-Aware Hierarchical Attention Recurrent Neural Networks for Multi-Criteria Recommender System

Manogna Vankayalapati[1], V Ramanjaneyulu Yannam[2], Sarada Korrapati[3], Murali Krishna Enduri[4]*
Department of CSE, SRM University, AP, India[1,3,4]
Department of CSE, ABV-IIITM, Gwalior, India[2]

***Abstract*—Recommendation systems are an important component for various online platforms, especially in the e-commerce domain. Recommendation systems suggest items to users using information from their past interactions such as reviews, ratings, and purchase history. Traditional recommendation systems allow users to give only a single rating for an item. Recently, deep learning approaches have been used to improve recommendation accuracy in single rating systems, but these systems do not provide enough information about user preferences for an item. Domains such as gaming, movies, and tourism enable users to give ratings on multiple criteria for an item, which makes it easier to understand user preferences compared to single rating systems. In this study, we propose a Time-Aware Hierarchical Attention Recurrent Neural Network (TAH-RNN), a deep learning-based approach designed to utilize ratings from multiple criteria. Our proposed approach helps understand the association between multiple criteria ratings and overall ratings for each user. The model integrates temporal dynamics with multi-criteria ratings by applying a Time-Aware Importance-Based Sequence Formation mechanism, which assigns importance weights to each criterion based on interaction time and enables hierarchical attention to learn their relationships over sequential user behavior. Experiments using real-world datasets (TripAdvisor, BeerAdvocate, and Skytrax Airlines) indicate that the proposed approach performs well compared to single rating systems and multiple criteria approaches across various metrics.**

Keywords*—*Recommendation system; multi-criteria ratings; time-aware; hierarchical attention; recurrent neural networks; user preferences

I. Introduction

In recent times, recommendation systems (RS) have become an important part of digital platforms, especially in the e-commerce domain, to handle large volumes of data and assist users in finding relevant items. The increase in digital content and the growing dependency on digital services have increased the demand for recommendation systems that can effectively filter large amounts of information for users. The main objective of a recommendation systems is generating personalized suggestions by analyzing users preferences by utilizing the information available on online platforms such as ratings, reviews, and purchase histories. The personalized suggestions reduce user effort in navigating through vast item collections and help businesses improve user engagement and satisfaction. This process of generation of suggestions is performed in two ways: content-based filtering and collaborative filtering (CF). Content-based filtering methods suggest items to users by examining the features of current items and those previously preferred by users [23], [21]. CF techniques are considered the most effective and commonly used in the e-commerce domain. These algorithms suggest items that have not yet been rated by a user by leveraging the preferences and rating patterns of other users with similar interests. CF methods are grouped into two categories: Memory-based Collaborative Filtering and Model-based Collaborative Filtering. Memory-based Collaborative Filtering produces recommendations by utilizing the complete user-item rating data and identifying similar user (neighbors) of the target user, then using their ratings to generate suggestions [39]. In Memory-based Collaborative Filtering, it is further classified into two categories: User-based Collaborative Filtering and Item-based Collaborative Filtering. In User-based Collaborative Filtering, users with similar rating patterns are identified, and items preferred by these users are suggested to the active user [24]. Item-based Collaborative Filtering measures similarities between items and recommends items similar to those already rated by the active user [37]. Both approaches are widely used in online platforms due to their effectiveness in capturing similarities from the rating data. Model-based Collaborative Filtering on the other hand, applies machine learning or statistical methods to learn patterns from user-item rating data and estimates ratings for items that have not yet been evaluated by the user. This approach often achieves improved accuracy by learning latent representations of users and items during model training [24], [19]. There are several other efficient approaches namely, matrix factorization and deep neural networks are used to model user preferences [13], [15], [16], [17], [8], [30], [25], [26]. The above methods utilize the traditional single rating systems.

However, single rating systems fail to capture user opinions across different aspects or features, especially in service areas such as hotels, movies, games, etc. To address the limitation of single rating systems, multiple criteria recommender systems are used by enabling users to provide ratings across multiple aspects of an item, such as service, quality, or value. For example, a user can give five ratings, which include the overall rating for a movie, with the criterion such as acting, story, direction, and visuals. Thus, multi-criteria recommender systems provide more comprehensive understanding of user preferences and enhance the personalization of recommendations compared to traditional single-rating systems [14], [20], [22], [12]. Unlike single-criterion recommendation systems, where temporal modeling focuses on the evolution of a single rating over time, multi-criteria recommender systems introduce additional complexity by requiring the modeling of temporal dynamics across multiple criteria. Each criterion may evolve differently over time based on user preferences. Therefore, it is necessary to jointly capture both temporal dependencies within each criterion and the relationships among multiple criteria,

*Corresponding author.

which requires more advanced modeling approaches compared to single-criterion systems.

Researchers G Adomavicius and Y Kwon [14], Lakiotaki et al. [20], Manouselis and Costopoulou [22] have explored the usage of multi-criteria rating information to improve the recommendations. However, machine learning approaches such as ensemble learning and reinforcement learning remain unexplored in the designing of multi-criteria recommender systems. This study presents a deep learning-based approach called Time-Aware Hierarchical Attention RNN to effectively handle multi-criteria recommender systems. The model incorporates a time-aware importance-based sequence formation to adjust the significance of each criterion based on the frequency of users' recent interactions, and employs hierarchical attention to capture both intra-criteria and inter-criteria relationships. In addition, a RNN module is integrated to preserve long-term user preferences and prevent information loss over time. These improvements enable the model to learn complex dependencies between multiple criteria and overall ratings more effectively. The proposed approach outperforms various deep learning and existing approaches of recommender systems.

II. Related Work

Existing research on recommender systems can be generally categorized into multi-criteria models, sequential models, hierarchical attention-based models, and temporal models.

A. Multi-Criteria Models

In early research on multi-criteria recommender systems, Lakiotaki et al. [20] proposed a user modeling framework that considered multiple aspects of user feedback to improve recommendations. This approach enabled a more comprehensive understanding of user preferences across different dimensions, such as quality, service, and value. This model faced limitations in scalability and showed poor adaptability to the eveolving nature of user interest. Campos et al. [18] showed that a number of previously proposed time-aware multi-criteria recommendation models had previously ignored some very important factors, which reduced the performance of the prediction model. Bougteb et al. [12] extended the previously proposed models by introducing the Multi-Criteria Attention-LSTM model, which uses attention mechanisms to handle ratings based on multiple criteria. Although the model showed promise in terms of performance and the privacy of the users, the model showed poor performance in terms of the computational cost and the sensitivity of the model. Other multi-criteria-based models, such as Nassar et al. [28], showed the effectiveness of deep learning in multi-criteria recommendation scenarios, which still did not address the temporal dimension of the model.

B. Sequential Models

With the introduction of deep learning, recurrent neural networks became prominent in recommendation system. Xu et al. [9] introduced the Long and Short-Term Self-Attention Network (LSSA), which demonstrates effectiveness in capturing the balance between short-term and long-term user preferences using a combination of recurrent and self-attention layers. Similarly, Hu et al. [4] proposed the Memory-Augmented Attention Network (MEANS), which used an external memory to retain historical user interactions for long-term dependency modeling. These RNN-based methods effectively capture sequential dynamics but require high computational cost and lack explicit hierarchical or criteria-level representations, which limit their interpretability and generalization to multi-criteria environments. Foundational sequential models such as GRU4Rec by Hidasi et al. [15] and attention-based sequential models such as SASRec proposed by Kang et al. [29], BERT4Rec by Sun et al. [31], and S^3-Rec by Zhou et al. [32] introduced self-attention and self-supervised learning for stronger sequence modeling. Oh et al. [7] improved sequential modeling using multimodal attention, Yuan et al. [10] incorporated contextual signals into GRU-based architectures, and Xu et al. [33] combined graph structures with self-attention.

C. Hierarchical Attention-Based Models

With improvements in attention mechanisms, researchers began exploring hierarchical and contextual attention-based models to capture more complex dependencies. Cui et al. [5] introduced the Hierarchical Contextual Attention-Based GRU (HCA-GRU), which applied two levels of attention to aggregate both local and global contextual information. Ruan et al. [6] expanded on this idea by proposing the Knowledge-Enhanced Personalized Hierarchical Attention Network (KEPHAN), which uses external knowledge graphs to improve personalization and better understand what items represent. Elsayed et al. [2] introduced the Hierarchical Masked Attention for Multi-Behaviour Recommendation (HMAR), which added another approach to this domain. Other attention-based models, like the DIN model proposed by Zhou et al. [36], have also shown strong performance in capturing users' interests. Even though these hierarchical models provided better interpretability and personalization, they were still computationally expensive and required complete contextual or semantic information available.

D. Temporal Models

In addition, time-aware recommendation systems were also proposed, which considered time as a factor by capturing the dynamics of users' interests. Wang et al. [3] introduced a Time-Aware Sequence Model (IDLSTM-EC), that integrates interval and duration gating mechanism using an LSTM architecture. Zhang et al. [11] proposed a Time-Aware Self-Attention Neural Network (TAT4SREC), which incorporate time intervals as a bias terms in attention weight calculations. Following this direction, Zhang et al. [1] proposed a Hierarchical Time-Aware Mixture of Experts (HM4SR). Jain et al. [27] proposed a time-aware model using Gower's coefficient and time decay functions. Wide & Deep [34] and DeepFM [35] proposed by Guo et al. also contributed to recommendation systems using deep feature-based models. Time-aware recommendation models were found effective in understanding users' behavior over time and improving recommendation accuracy. However, they require a lot of data and computational resources, making their practical usage difficult.

Despite these advancements, existing approaches address multi-criteria modeling, sequential modeling, hierarchical dependencies, and temporal modeling independently. However, they fail to jointly integrate these aspects into a unified framework, which limits their ability to capture the evolving multi-criteria user preferences [5], [29], [28].

III. Background

Recommender systems are an integral component of several digital platforms, including e-commerce, travel, and streaming services. The main idea behind the recommendation system is to assist users in finding items that match their preferences by selecting the most suitable options from the large set of available items. The traditional recommender system mainly focused on collaborative filtering, with specific emphasis on matrix factorization, for revealing the underlying relationships between users and items [13]. Though the traditional system performed well, it mainly considered user preferences and item characteristics as static, which does not reflect the realistic scenarios in which user preferences change with time. In addition, many applications collect ratings on multiple aspects (such as value, service, and cleanliness in hotel reviews). Predicting a single overall score fails to include the complete details of user feedback [14].

Deep learning models have been adapted to address the limitations of static recommendation methods by capturing sequential and time-aware user behaviors. Recurrent neural networks (RNNs) are effective in modeling session-based interactions by analyzing the sequence of user actions, including the order of viewed items [15]. Recurrent recommender networks integrate the temporal dynamics of users and items, leading to higher accuracy and personalization in predictions [16]. Recently, attention mechanisms have significantly impacted sequence modeling, with the Transformer showing that self-attention can efficiently capture long-term dependencies [17].

IV. Proposed Approach

We propose a Time-Aware Hierarchical Attention Recurrent Neural Network for multi-criteria recommender systems (see Fig. 1). The proposed approach combines various components such as embedding process, time-aware importance-based sequence formation, hierarchical attention, RNN, and rating prediction. By combining these components, the model captures user behavior over time, dependencies between different criteria, and long-term patterns. These components are applied to three different datasets, namely TripAdvisor, BeerAdvocate, and Skytrax Airlines.

A. Embedding Process

In the embedding process, each user, item, and criterion is mapped into a continuous latent space that stores semantic interactions. The corresponding embedding vectors are denoted as, $\mathbf{E}_u \in \mathbb{R}^{N_u \times d}$, $\mathbf{E}_i \in \mathbb{R}^{N_i \times d}$, and $\mathbf{E}_c \in \mathbb{R}^{C \times d}$, where, N_u and N_i represent the total number of users and items, respectively, C denotes the number of criteria, and d is an embedding dimension. For each interaction occurring at timestamp t, the latent embeddings for user, item, and criterion k are defined as $\mathbf{e}_u = \mathbf{E}_u[u_t]$, $\mathbf{e}_i = \mathbf{E}_i[i_t]$, and $\mathbf{e}_c^{(k)} = \mathbf{E}_c[k]$, respectively. Each criterion embedding is normalized by rating scale $\frac{r_t^{(k)}}{5.0}$ to preserve the rating within the embedding representation. Thus, the criterion-wise weighted representation is formulated as:

$$\mathbf{z}_t^{(k)} = \mathbf{e}_c^{(k)} \cdot \frac{r_t^{(k)}}{5.0}, \quad k = 1, 2, \ldots, C \tag{1}$$

This process helps to combine the user ID, item ID, and each criterion rating into a unified latent representation. By doing this, the network can understand the complex interactions among users, items, and the criteria.

B. Time-Aware Importance-Based Sequence Formation

The embedding vectors computed in Eq. (1) are used to measure how recent user–item interactions are, by giving more importance to recent interactions compared to older ones. For each criterion k, two parameters are used: the decay rate $\lambda^{(k)}$ and the importance weight $\gamma^{(k)}$. The time gap from the first interaction is defined as $\Delta_t = \tau_t - \tau_0$, where τ_0 denotes the timestamp of the initial interaction. The time-aware importance score is defined as Eq. (2):

$$\beta_t^{(k)} = \gamma^{(k)} \cdot \exp\Big(-\,\mathrm{softplus}(\lambda^{(k)}) \cdot \Delta_t\Big) \tag{2}$$

where, the softplus activation function is used to ensure that the decay rate remains positive by stating that $\lambda^{(k)} \geq 0$. After this softmax activation function is applied throughout time [see Eq. (3)].

$$\alpha_t^{(k)} = \frac{\exp(\beta_t^{(k)})}{\sum_{j=1}^{T} \exp(\beta_j^{(k)})} \tag{3}$$

Finally, the criterion feature vector with temporal information is computed as:

$$\mathbf{h}_t^{(k)} = \alpha_t^{(k)} \cdot \mathbf{z}_t^{(k)} \tag{4}$$

This process gives more importance to recent user behavior by reducing the older interactions and focusing on the most relevant ones.

C. Hierarchical Attention Mechanism

The time-aware representations $\mathbf{h}_t^{(k)}$ obtained from Eq. (4) are organized into a criterion-specific sequence $\mathbf{H}^{(k)} = [\,\mathbf{h}_1^{(k)}, \mathbf{h}_2^{(k)}, \ldots, \mathbf{h}_T^{(k)}\,]$, which serves as input to the hierarchical attention mechanism defined in Eq. (5) to Eq. (10). This enables the model to capture both intra-criterion (micro-level) and inter-criterion (macro-level) dependencies, allowing a better understanding of user preferences across different criteria.

Each vector is projected into query, key, and value spaces as:

$$\mathbf{Q}^{(k)} = \mathbf{H}^{(k)}\mathbf{W}_Q^{(k)}, \quad \mathbf{K}^{(k)} = \mathbf{H}^{(k)}\mathbf{W}_K^{(k)}, \quad \mathbf{V}^{(k)} = \mathbf{H}^{(k)}\mathbf{W}_V^{(k)} \tag{5}$$

where, $\mathbf{W}_Q^{(k)}, \mathbf{W}_K^{(k)}, \mathbf{W}_V^{(k)} \in \mathbb{R}^{d \times d}$ are learnable matrices. The attention output can be obtained from:

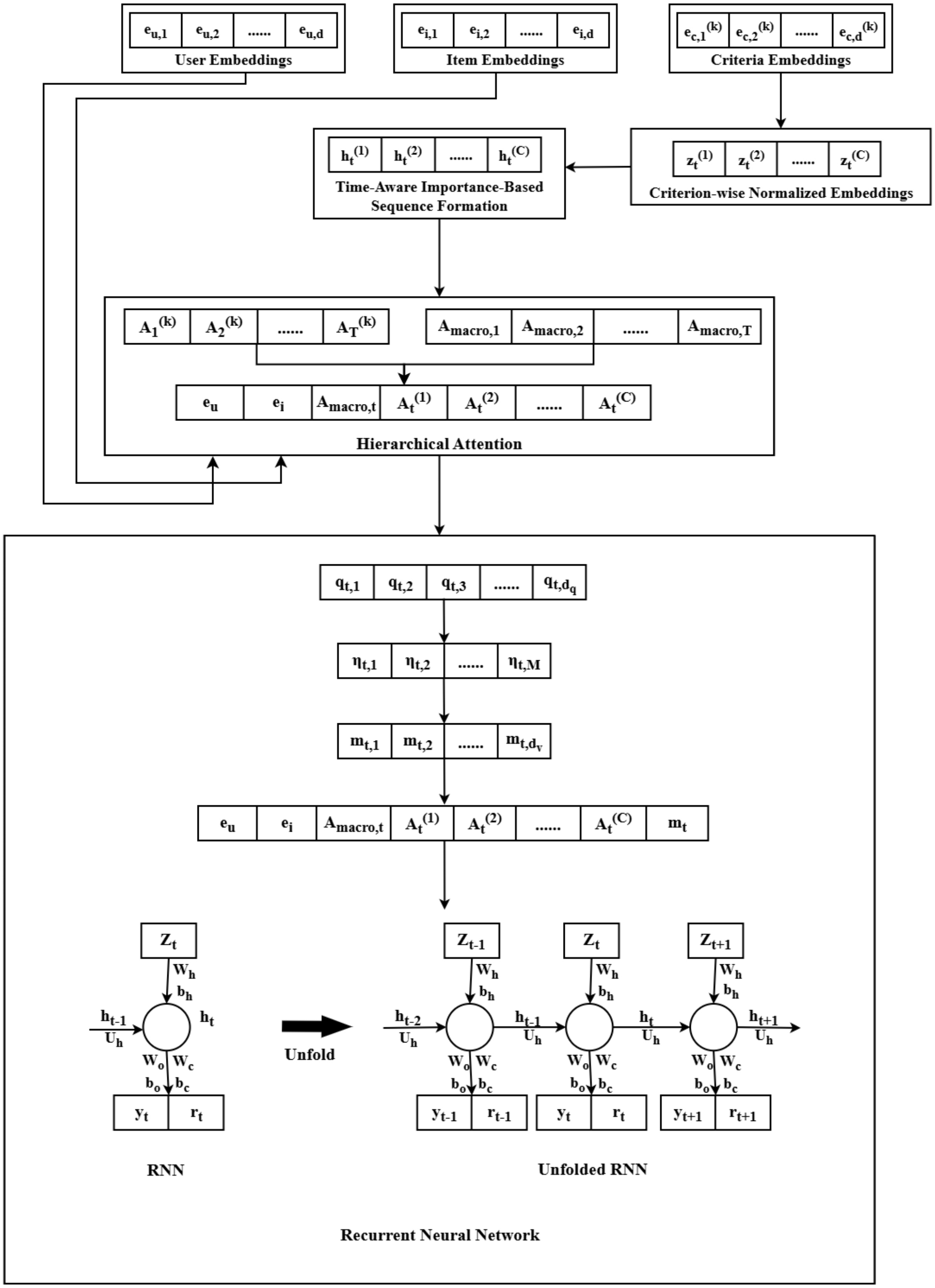

Fig. 1. Proposed architecture of Time-Aware Hierarchical Attention Recurrent Neural Network model.

$$\mathbf{A}^{(k)} = \mathrm{softmax}\left(\frac{\mathbf{Q}^{(k)}\mathbf{K}^{(k)\top}}{\sqrt{d}}\right)\mathbf{V}^{(k)} \tag{6}$$

This helps the model focus on the most relevant time steps for each criterion.

To capture dependencies among multiple criteria, outputs from all micro-level representations are integrated. The aggregated representation is computed as:

$$\mathbf{H}_{\text{macro}} = \frac{1}{C}\sum_{k=1}^{C}\mathbf{A}^{(k)} \tag{7}$$

The resulting macro-level output can be represented as: $\mathbf{H}_{\text{macro}} = [\,\mathbf{h}_{\text{macro},1},\, \mathbf{h}_{\text{macro},2},\, \ldots,\, \mathbf{h}_{\text{macro},T}\,]$

Then, query, key, and value projections are defined as:

$$\begin{aligned}\mathbf{Q}_{\text{macro}} &= \mathbf{H}_{\text{macro}}\mathbf{W}_Q\\ \mathbf{K}_{\text{macro}} &= \mathbf{H}_{\text{macro}}\mathbf{W}_K\\ \mathbf{V}_{\text{macro}} &= \mathbf{H}_{\text{macro}}\mathbf{W}_V\end{aligned} \tag{8}$$

and the macro-level output is obtained by:

$$\mathbf{A}_{\text{macro}} = \mathrm{softmax}\left(\frac{\mathbf{Q}_{\text{macro}}\mathbf{K}_{\text{macro}}^{\top}}{\sqrt{d}}\right)\mathbf{V}_{\text{macro}} \tag{9}$$

The final hierarchical attention representation integrates both levels:

$$\mathbf{X}_t = [\,\mathbf{e}_u \| \mathbf{e}_i \| \mathbf{A}_{\text{macro},t} \| \mathbf{A}_t^{(1)} \| \ldots \| \mathbf{A}_t^{(C)}\,] \tag{10}$$

This combination ensures that both local intra-criterion (micro-level) and inter-criterion (macro-level) are jointly encoded.

D. Recurrent Neural Network

To accurately model long-term dependencies that extend beyond sequential interactions, a **Recurrent Neural Network (RNN)** is incorporated into the proposed architecture. The memory consists of learnable key–value matrices $\mathbf{M}_K \in \mathbb{R}^{M\times d}$ and $\mathbf{M}_V \in \mathbb{R}^{M\times d}$, where M is the number of memory slots and d denotes the dimensionality of the embeddings. The hierarchical contextual representation $\mathbf{X}_t$ obtained after the attention mechanisms is aggregated over time to construct a memory query vector as Eq. (11):

$$\mathbf{q}_t = \mathbf{W}_q\left(\frac{1}{T}\sum_{j=1}^{T}\mathbf{X}_j\right) \tag{11}$$

where, $\mathbf{W}_q \in \mathbb{R}^{d\times d}$ is a learnable projection matrix. The attention weights over the memory keys are then computed using a softmax function, Eq. (12):

$$\boldsymbol{\eta}_t = \mathrm{softmax}\big(\mathbf{q}_t\mathbf{M}_K^{\top}\big) \tag{12}$$

where, $\boldsymbol{\eta}_t \in \mathbb{R}^M$ represents the normalized importance distribution over the M memory slots. The corresponding memory read vector is calculated as a weighted sum of the value embedding [see Eq. (13)]:

$$\mathbf{m}_t = \boldsymbol{\eta}_t\mathbf{M}_V \tag{13}$$

which helps retrieve relevant contextual information from memory. The resulting memory vector is then combined with the hierarchical embedding to form the input to the RNN as shown below, Eq. (14):

$$\mathbf{Z}_t = [\,\mathbf{X}_t \| \mathbf{m}_t\,] \tag{14}$$

where, $\|$ denotes vector concatenation. The Recurrent Neural Network then updates its hidden state using a non-linear transformation [see Eq. (15)]:

$$\mathbf{h}_t = \tanh(\mathbf{W}_h\mathbf{Z}_t + \mathbf{U}_h\mathbf{h}_{t-1} + \mathbf{b}_h) \tag{15}$$

where, $\mathbf{b}_h$ represents the bias vector and $\mathbf{W}_h, \mathbf{U}_h \in \mathbb{R}^{d\times d}$ are trainable vectors. This module enables the model to dynamically retrieve relevant information from memory while preserving the sequence of interactions, which improves the representation of temporal and contextual dependencies in user-item interactions.

E. Evaluation Metrics

The final hidden state $\mathbf{h}_t$ obtained from the Recurrent Neural Network is passed through two different fully connected layers for estimate the overall rating and the multi-criteria ratings. The formulation for predicting the overall rating is as follows [see Eq. (16)]:

$$\hat{y}_t = \mathbf{W}_o\mathbf{h}_t + b_o \tag{16}$$

The multi-criteria rating prediction is formulated as follows, Eq. (17):

$$\hat{\mathbf{r}}_t = \mathbf{W}_c\mathbf{h}_t + \mathbf{b}_c \tag{17}$$

The training of this model is done by minimizing the Mean Squared Error, which evaluates the difference between predicted ratings and true ratings for each criterion, as shown in the following Eq. (18):

$$\mathcal{L}_{MSE} = \frac{1}{NC}\sum_{i=1}^{N}\sum_{k=1}^{C}(r_{i,k} - \hat{r}_{i,k})^2 \tag{18}$$

Let N be the total count of the training samples, and C be the total count of criteria ratings. The symbol $r_{i,k}$ denotes the actual rating of the i-th sample for the k-th criterion, whereas $\hat{r}_{i,k}$ denotes the predicted rating. The mean squared error (MSE) measures the average squared difference between the predicted and actual ratings.

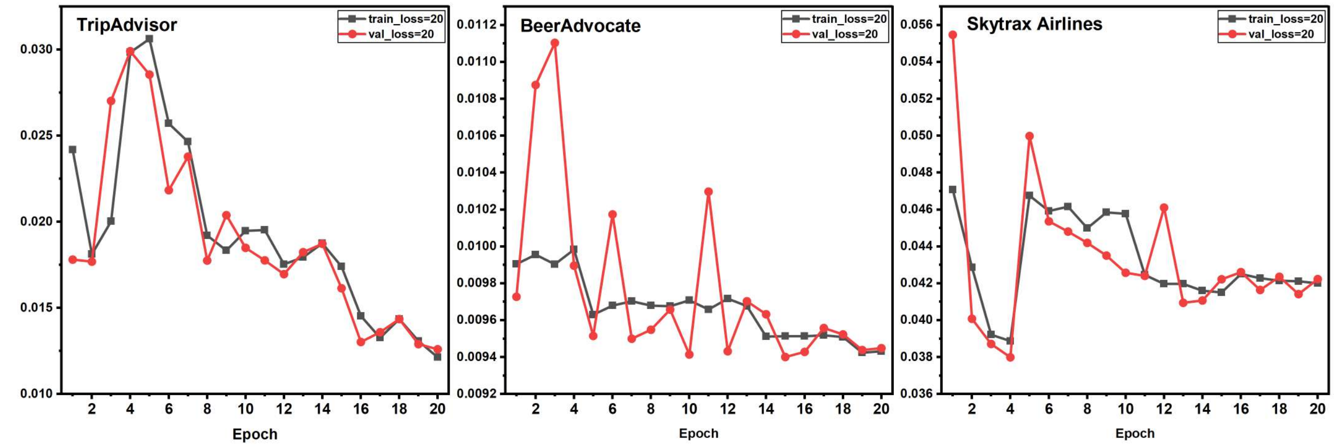

Fig. 2. Comparison of loss on different datasets for proposed model.

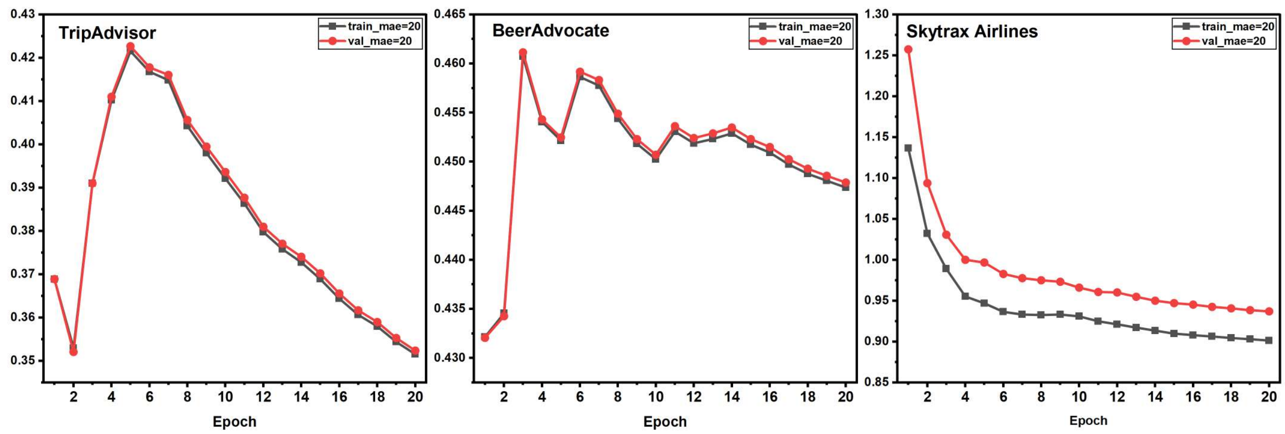

Fig. 3. Comparison of MAE on different datasets for proposed model.

To evaluate the model performance in generating recommendations, the mean absolute error (MAE) and the root mean squared error (RMSE) are calculated as follows [see Eq. (19) and Eq. (20)]:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{19}$$

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{20}$$

The evaluation metrics are computed for the overall rating to thoroughly assess the model's performance across various dimensions of user feedback.

V. Dataset Description

To evaluate the TAH-RNN model, three real-world datasets with multi-criteria rating were utilized: TripAdvisor, BeerAdvocate, and Skytrax Airlines. The TripAdvisor dataset used in this research is based on the HotelRec corpus provided by Diego Antognini and Boi Faltings, which contains around 50 million hotel reviews. Out of this vast amount of data, a sample of 100,000 user-item interactions has been selected for the experimental process. The BeerAdvocate dataset was taken from Kaggle, and it includes nearly 1 million reviews; 100K samples were used for this work, and Skytrax Airlines dataset was also sourced from Kaggle and contains 65,947 records after cleaning. For all datasets, the only necessary features such as user id, item id, timestamp, overall rating, and the available criteria ratings were included. The detailed summary of all three datasets are presented in Table I.

TABLE I. Summary of Datasets

Dataset	Users	Items	Criteria	Overall Rating	Criteria Rating	Size
TripAdvisor [38]	96,874	62,210	8	1–5	1–5	100,000
BeerAdvocate [40]	12,371	19,053	4	1–5	1–5	100,000
Skytrax Airlines [41]	44,069	81	6	1–10	1–5	65,947

VI. Experimental Setup

The model was trained by utilizing the AdamW optimizer with a batch size of 32 alogn with embedding dimension of 64 for all datasets. The number of training epochs was set to 20, and a weight decay 1×10^{-5} was applied to prevent overfitting. The learning rate was adjusted for each dataset (0.01 for TripAdvisor, 0.02 for BeerAdvocate, and 0.03 for Skytrax Airlines) to account for differences in dataset scale and rating distributions. This dataset-specific tuning ensured stable convergence and improved model performance while maintaining consistent training settings across all experiments. All baseline models were implemented and evaluated in the same experimental environment using identical preprocessing steps, dataset partitioning, and evaluation measures to mantain a fair and consistent comparison.

VII. Results and Discussion

In this section, we discuss about MAE and RMSE evaluation results performed on TripAdvisor, BeerAdvocate, and Skytrax Airlines datasets using TAH-RNN (Time-Aware Hierarchical Attention Recurrent Neural Networks). The model was trained under 20 epochs, and its performance was evaluated at each epoch to observe improvements during training. MAE and RMSE were calculated using the difference between predicted and actual ratings during both the training and validation phases.

A. Performance Comparison on Different Datasets

From Fig. 2, the training and validation loss values observed across the TripAdvisor, BeerAdvocate, and Skytrax Airlines datasets indicate that the TAH-RNN model converges effectively during training. The overall decreasing trend in loss, along with the minimal difference between training and validation values, demonstrates that the model learns meaningful patterns without overfitting. The TripAdvisor dataset demonstrates stable learning behavior due to relatively structured user interactions, while the BeerAdvocate dataset shows consistent performance, indicating strong generalization. Although the Skytrax Airlines dataset exhibits more variability due to diverse user ratings, the model maintains a steady improvement in loss, highlighting its robustness across different data characteristics.

From Fig. 3, the MAE trends for the TripAdvisor, BeerAdvocate, and Skytrax Airlines datasets demonstrate that the TAH-RNN model provides accurate and consistent predictions across different domains. The minimal difference between training and validation MAE values indicates that the model generalizes well and avoids overfitting. The TripAdvisor and BeerAdvocate datasets show stable and consistent behavior, reflecting the model's ability to understand user preferences effectively. In contrast, the Skytrax Airlines dataset presents more variability due to complex and diverse user feedback; however, the model maintains consistent performance, demonstrating its capability to handle heterogeneous data.

From Fig. 4, the RMSE trends further confirm the effectiveness of the TAH-RNN model in capturing prediction accuracy across all datasets. The minimal difference between training and validation RMSE indicates consistent performance without overfitting. The TripAdvisor dataset shows consistent behavior, while the BeerAdvocate dataset maintains stable trends, reflecting reliable prediction performance. Although the Skytrax Airlines dataset exhibits higher variability due to diverse rating patterns, the model continues to perform consistently, highlighting its robustness in handling complex multi-criteria data. Overall, the RMSE trends validate the model's ability to effectively capture both temporal patterns and hierarchical dependencies.

B. Performance Comparison on Different Models

The proposed model performed better than the baseline models on the TripAdvisor dataset, as shown in Table II. The comparison includes models such as KNN, MC-ANN, LSTM, and LSTM-RNN. The TAH-RNN model achieved the lowest MAE and RMSE across all methods. This improvement is due to the ability to capture temporal dynamics and hierarchical relationships across multiple criteria, which helps in learning user preferences more effectively.

TABLE II. Results for TripAdvisor Dataset

Model	MAE	RMSE
KNN	0.9659	1.3568
MC-ANN	0.6947	0.8871
LSTM	0.5500	0.7841
LSTM-RNN	0.5675	0.7903
TAH-RNN	**0.3521**	**0.5401**

TABLE III. Results for BeerAdvocate Dataset

Model	MAE	RMSE
KNN	0.4911	0.6525
MC-ANN	0.6208	0.8057
LSTM	0.4597	0.5998
LSTM-RNN	0.4806	0.6259
TAH-RNN	**0.4320**	**0.5903**

The proposed model outperformed baseline models on the BeerAdvocate dataset, such as KNN, MC-ANN, LSTM, and LSTM-RNN, as shown in Table III. The TAH-RNN model obtained the lowest MAE and RMSE values. Although the performance difference is smaller compared to the TripAdvisor dataset, the model still demonstrated better accuracy by effectively capturing temporal variations and multi-criteria dependencies in user preferences.

TABLE IV. Results for Skytrax Airlines Dataset

Model	MAE	RMSE
KNN	3.1330	4.1276
MC-ANN	2.3294	2.7109
LSTM	1.3830	1.7157
LSTM-RNN	1.6899	2.0432
TAH-RNN	**0.9369**	**1.1522**

The proposed model significantly outperformed baseline models on the Skytrax Airlines dataset, such as KNN, MC-ANN, LSTM, and LSTM-RNN, as shown in Table IV. The TAH-RNN model achieved lower MAE and RMSE when compared to other models. This highlights the effectiveness of the proposed approach in handling large and complex multi-criteria datasets through the integration temporal dynamics and hierarchical attention.

The consistent performance improvements across all three datasets indicate the generalization ability of the proposed TAH-RNN model.

The improved performance of the proposed TAH-RNN model is due to its ability to capture temporal variations in user preferences. In real-world scenarios, user interests

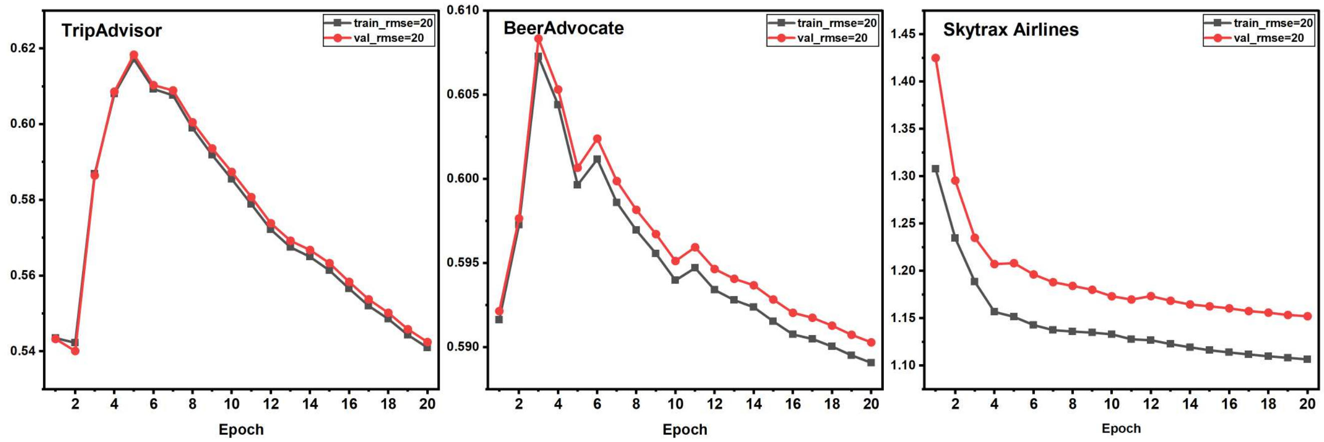

Fig. 4. Comparison of RMSE on different datasets for proposed model.

change over time, and recent interactions generally provide a better indication of current preferences than older ones. The time-aware importance-based sequence formation mechanism assigns higher weights to recent interactions while reducing the influence of outdated information, which helps the model focus on more relevant patterns. Additionally, modeling temporal behavior at the criterion level enables the system to capture fine-grained changes in user preferences across multiple aspects, leading to improved recommendation accuracy.

Despite its effectiveness, the proposed model has certain limitations. First, the assumption that recent interactions are always more important may not hold for users with stable long-term preferences. Second, the model requires sufficient historical interaction data to learn meaningful temporal patterns, making it less effective in cold-start scenarios. Third, the combination of hierarchical attention and temporal modeling increases computational complexity, which may result in longer training time. Furthermore, the model may struggle to capture sudden changes in user preferences that do not follow gradual temporal trends.

VIII. Conclusion

In this study, a Time-Aware Hierarchical Attention Recurrent Neural Network (TAH-RNN) model is proposed for handling multi-criteria ratings. The proposed approach combines multi-criteria ratings with temporal information at the input layer to capture user behavior. The experimental results obtained from real-world datasets demonstrates that the proposed approach performs well and provides scope for further exploration. However, the proposed model has certain limitations, including its dependency on sufficient historical interaction data and its computational complexity due to the integration of temporal modeling and hierarchical attention. In addition, RNN-based architectures may have limitations in capturing long-term dependencies effectively, which can affect performance in complex sequential scenarios. Further improvements can be achieved by exploring advanced recurrent architectures such as LSTM and GRU variants to better capture temporal dependencies and improve learning efficiency. Additionally, Transformer-based models can be explored to better handle long-term dependencies and improve scalability. Furthermore, integrating user reviews along with multi-criteria ratings can enhance the understanding of user preferences and further improve recommendation quality.

References

[1] S. Zhang, L. Chen, D. Shen, C. Wang, and H. Xiong. Hierarchical Time-Aware Mixture of Experts for Multi-Modal Sequential Recommendation. In *Proceedings of the ACM Web Conference (WWW '25)*, pages 3672–3682, 2025. doi:10.1145/3696410.3714676.

[2] S. Elsayed, A. Rashed, and L. Schmidt-Thieme. HMAR: Hierarchical Masked Attention for Multi-Behaviour Recommendation. *arXiv preprint arXiv:2405.09638*, 2024. doi:10.48550/arXiv.2405.09638.

[3] D. Wang, D. Xu, and G. Xu. Time-Aware sequence model for next-Item recommendation. *Applied Intelligence*, 51:906–920, 2021. doi:10.1007/s10489-020-01820-2.

[4] C. Hu, P. He, C. Sha, and J. Niu. MEANS: Memory-Augmented Attention Network for Sequential Recommendation. In *Web Information Systems Engineering (WISE) 2019*, pages 228–242, 2019. doi:10.1007/978-3-030-34223-4_15.

[5] Q. Cui, S. Wu, Y. Huang, and L. Wang. A hierarchical contextual attention-based network for sequential recommendation. *Neurocomputing*, 358:141–149, 2019. doi:10.1016/j.neucom.2019.04.073.

[6] S. Ruan, C. Yang, and D. Li. Knowledge-Enhanced Personalized Hierarchical Attention Network for Sequential Recommendation. *World Wide Web*, 27(2), 2024. doi:10.1007/s11280-024-01236-9.

[7] H. Oha, W. Joa, and D. Kim. Attention-Based Sequential Recommendation System Using Multimodal Data. *arXiv preprint arXiv:2405.17959*, 2024. doi:10.48550/arXiv.2405.17959.

[8] P. Wei, H. Shu, J. Gan, X. Deng, Y. Liu, W. Sun, T. Chen, C. Hu, Z. Hu, Y.Deng, W. Qin, Z Li. Sequential Recommendation System Based on Deep Learning: A Survey. *Electronics*, 14(11), 2025. doi:10.3390/electronics14112134.

[9] C. Xu, J. Feng, P. Zhao, F. Zhuang, D. Wang, Y. Liu, and V. S. S. Yanchi Liu. Long- and Short-Term Self-Attention Network for Sequential Recommendation. *Neurocomputing*, 423:580–589, 2021. doi:10.1016/j.neucom.2020.10.066.

[10] W. Yuan, H. Wang, X. Yu, N. Liu, Z. Li. Attention-Based Context-Aware Sequential Recommendation Model. *Information Sciences*, 510:122–134, 2020. doi:10.1016/j.ins.2019.09.007.

[11] Y. Zhang, B. Yang, H. Liu, D. Li. A time-aware self-attention based neural network model for sequential recommendation. *Applied Soft Computing*, 133, 2023. doi:10.1016/j.asoc.2022.109894.

[12] Y. Bougteb, B. Ouhbi, B. Frikh, E. Zemmouri. A Multi-Criteria Attention-LSTM Approach for Enhancing Privacy and Accuracy in Recommender Systems. *Social Network Analysis and Mining*, 15:38, 2025. doi:10.1007/s13278-025-01458-3.

[13] Y. Koren, R. Bell, C. Volinsky. Matrix Factorization Techniques for Recommender Systems. *Computer*, 42(8):30–37, 2009. doi:10.1109/MC.2009.263.

[14] G. Adomavicius, Y. Kwon. Multi-criteria Recommender Systems. In *Recommender Systems Handbook*, pages 847–880. Springer, 2015. doi:10.1007/978-1-4899-7637-6_25.

[15] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk. Session-based Recommendations with Recurrent Neural Networks. In *Proceedings of ICLR*, 2016. doi:10.48550/arXiv.1511.06939.

[16] C.-Y. Wu, A. Ahmed, A. Beutel, A. J. Smola, and H. Jing. Recurrent Recommender Networks. In *Proceedings of KDD*, pages 495–503, 2017. doi:10.1145/3018661.3018689.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Łukasz Kaiser, I. Polosukhin. Attention Is All You Need. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5998–6008, 2017. doi:10.48550/arXiv.1706.03762.

[18] P. G. Campos, F. Díez, I. Cantador. Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols. *User Modeling and User-Adapted Interaction*, 24:67–119, 2014. doi:10.1007/s11257-012-9136-x.

[19] J. L. Herlocker, J. A. Konstan, L. G. Terveen, J. T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22(1):5–53, 2004. doi:10.1145/963770.963772.

[20] K. Lakiotaki, N. F. Matsatsinis, A. Tsoukiàs. Multi-criteria user modeling in recommender systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 42(3):421–431, 2011. doi:10.1109/mis.2011.33.

[21] P. Lops, M. de Gemmis, G. Semeraro. Content-based recommender systems: State of the art and trends. In *Recommender Systems Handbook*, pages 73–105. Springer, 2011. doi:10.1007/978-0-387-85820-3_3.

[22] N. Manouselis, C. Costopoulou. Analysis and classification of multi-criteria recommender systems. *World Wide Web*, 10:415–441, 2007. doi:10.1007/s11280-007-0019-8.

[23] M. J. Pazzani, D. Billsus. Content-based recommendation systems. In *The Adaptive Web*, pages 325–341. Springer, 2007. doi:10.1007/978-3-540-72079-9_10.

[24] B. Sarwar, G. Karypis, J. Konstan, J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of WWW*, pages 285–295, 2001. doi:10.1145/371920.372071.

[25] S. Sukhbaatar, A. Szlam, J. Weston, R. Fergus. End-to-end memory networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[26] Y. Zhang, G. Lai, M. Zhang, Y. Zhang, Y. Liu, S. Ma. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of SIGIR*, 2014. doi:10.1145/2600428.2609579.

[27] G. Jain, T. Maharab, A. Kumar, S. Sharma. Time-Aware Based Recommendation System using Gower's Coefficients: Enhancing Personalized Recommendation. *Procedia Computer Science*, 235:3379–3388, 2024. doi:10.1016/j.procs.2024.04.318.

[28] N. Nassar, A. Jafar, Y. Rahhal. Multi-criteria collaborative filtering recommender by fusing deep neural network and matrix factorization. *Journal of Big Data*, 7:34, 2020. doi:10.1186/s40537-020-00309-6.

[29] W.-C. Kang, J. McAuley. Self-Attentive Sequential Recommendation. In *2018 IEEE ICDM*, pages 197–206, 2018. doi:10.1109/ICDM.2018.00035.

[30] G. S. Balla, S. Tokala, and M. K. Enduri, "A Hybrid Recommender System using Louvain Community Detection in Complex Networks," in *Proceedings of the 2025 9th International Symposium on Innovative Approaches in Smart Technologies (ISAS)*, pp. 1–6, 2025, IEEE.

[31] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, P. Jiang. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *CIKM '19*, pages 1441–1450, 2019. doi:10.1145/3357384.3357895.

[32] K. Zhou, H. Wang, W. X. Zhao, Y. Zhu, S. Wang, F. Zhang, Z. Wang. S^3-Rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *CIKM '20*, pages 1893–1902, 2020. doi:10.1145/3340531.3411954.

[33] C. Xu, P. Zhao, Y. Liu, V. S. Sheng, J. X. F. Zhuang, J. Fang, X. Zhou. Graph contextualized self-attention network for session-based recommendation. In *Proceedings of IJCAI*, 2019. doi:10.24963/ijcai.2019/547.

[34] H.-T. Cheng L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir, R. Anil, Z. Haque, L. Hong, V. Jain, X. Liu, H. Shah. Wide & Deep Learning for Recommender Systems. In *DLRS 2016: Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, pages 7–10, 2016. doi:10.1145/2988450.2988454.

[35] H. Guo, R. Tang, Y. Ye, Z. Li, X. He, . DeepFM: A factorization-machine based neural network for CTR prediction. In *Proceedings of IJCAI*, pages 1725–1731, 2017. doi:10.48550/arXiv.1703.04247.

[36] G. Zhou, X. Zhu, C. Song, Y. Fan, H. Zhu, X. Ma, Y. Yan, J. Jin, H. Li, K. Gai. Deep interest network for click-through rate prediction. In *KDD '18*, pages 1059–1068, 2018. doi:10.1145/3219819.3219829.

[37] G. Linden, B. Smith, J. York. Amazon.com recommendations: item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80, 2003.

[38] D. Antognini, B. Faltings. HotelRec: a Novel Very Large-Scale Hotel Recommendation Dataset. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4917–4923, May 2020.

[39] D. Tallapally, R. S. Sreepada, B. K. Patra, and K. S. Babu. User Preference Learning in Multi-criteria Recommendations Using Stacked Auto Encoders. *RecSys '18: Proceedings of the 12th ACM Conference on Recommender System*, pp. 475-479, 2018. doi:10.1145/3240323.3240412

[40] Beer Reviews Dataset on Kaggle. https://www.kaggle.com/datasets/rdoume/beerreviews.

[41] Skytrax Airlines dataset. https://www.kaggle.com/datasets/efehandanisman/skytrax-airline-reviews.