

A Deep Learning Based Approach for Context-Aware Multi-Criteria Recommender Systems

Son-Lam VU and Quang-Hung LE*

Faculty of Information Technology, Quy Nhon University, Quy Nhon City, Vietnam

*Corresponding Author: Quang-Hung LE. Email: lequanghung@qnu.edu.vn

Received: 08 December 2021; Accepted: 14 January 2022

Abstract: Recommender systems are similar to an information filtering system that helps identify items that best satisfy the users' demands based on their preference profiles. Context-aware recommender systems (CARSS) and multi-criteria recommender systems (MCRSS) are extensions of traditional recommender systems. CARSS have integrated additional contextual information such as time, place, and so on for providing better recommendations. However, the majority of CARSS use ratings as a unique criterion for building communities. Meanwhile, MCRSS utilize user preferences in multiple criteria to better generate recommendations. Up to now, how to exploit context in MCRSS is still an open issue. This paper proposes a novel approach, which relies on deep learning for context-aware multi-criteria recommender systems. We apply deep neural network (DNN) models to predict the context-aware multi-criteria ratings and learn the aggregation function. We conduct experiments to evaluate the effect of this approach on the real-world dataset. A significant result is that our method outperforms other state-of-the-art methods for recommendation effectiveness.

Keywords: Recommender systems; context-aware; multi-criteria; deep learning; deep neural network

1 Introduction

A recommender system can be defined as a form of an information filtering system that is intended to provide items that could be of interest to the user [1]. Currently, along with the development and variety of products and services, recommender systems are increasingly widely used in areas such as online shopping (e.g., Amazon), e-news (e.g., Yahoo! News Today), music (e.g., Last.fm), travel (e.g., TripAdvisor), movies (e.g., Netflix), and social networks (e.g., Facebook). The traditional recommender system (also known as a two-dimensional recommender system) only uses two information dimensions about users and items, including user preferences for items (often reflected in ratings), user profiles and item content features to make recommendations.

CARSS are an extension of traditional recommender systems that give recommendations to users and consider contextual information (e.g., weather, time, and the user's mood) or latent contexts. In addition, user preference data are expanded into a multidimensional dataset, including users, items, and contextual



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

information [2]. Contextual information plays an important role and influences a user's item experience. The role of context has been recognized in improving the performance of a recommender system [3]. Another type of recommender system is MCRS, which utilizes user preferences in multiple criteria to better generate recommendations. For example, users on TripAdvisor.com can give ratings on multiple criteria (such as room size, cleanness, customer service, etc) in addition to the overall rating for a hotel.

Both CARS and MCRS have been widely applied in real-world applications, such as tourism, movies, music, and so on. Up to now, deep learning for recommender systems has gained significant results. However, how to exploit context in deep learning-based recommender systems is still an open issue. To our knowledge, there is no existing research that incorporates contextual information into deep learning-based MCRS. Although some recent related works [4,5] have exploited the methods to combine them. Zheng et al. [4] integrated context-awareness and multi-criteria decision making in the recommender systems by using the educational data as a case study. Meanwhile, a context-aware multi-criteria recommendation based on spectral graph partitioning was proposed by Dridi et al. [5]. Different from previous works, we aim to use deep learning for the context-aware multi-criteria recommendation. The main contributions in this work can be summarized as follows:

1. We propose a novel approach, which relies on deep learning for context-aware multi-criteria recommender systems. To the best of our knowledge, this is the first attempt to incorporate contextual information into MCRS using deep learning.
2. We apply DNN models to predict the context-aware multi-criteria ratings and learn the aggregation function.
3. We compare the proposed method with benchmark methods from the relevant state-of-art on the real-world dataset for recommendation effectiveness.

The rest of the paper is structured as follows. Section 2 gives some brief reviews of background knowledge. Our proposed method using deep learning for context-aware multi-criteria recommender systems is presented in Section 3. Section 4 analyses and discusses the evaluation results obtained. Related work is presented in Section 5. Finally, conclusions and future work are highlighted in Section 6.

2 Preliminaries

2.1 Recommender Systems

In systems that interact with users, such as e-commerce, e-news, and entertainment, users often express their interest in products or services by leaving comments, ratings (usually 1-5 stars), or information expressed through time/frequency of use. For example, a user leaves a 1-star review for a restaurant he/she has just visited, showing that he/she is not satisfied with this restaurant, or a user who watches a video often on the entertainment channel and for a long time shows that he/she is very interested in that video. The goal of the recommender system is to make recommendations about the items that users are most likely to be interested in. Typically, a recommender system works on two main entities, the user and the item. An important data source on which the recommender system can exploit information and make appropriate recommendations is historical data that shows the interaction between users and items. Typically, these data are organized in the form of a utility matrix composed of $Users \times Items$. In particular, each row in this matrix represents a user and his or her rating on different items, while each column represents an item and the ratings of different users on this item.

Formally, let us denote by U the set of users, and u is a user in U ; I is the set of items, i is an item in I , and Y is the rating matrix. We denote with r_{ui} the rating of user u for item i , and \hat{r}_{ui} presents the predicted rating of user u for item i . A recommender system attempts to estimate a rating function r such that

$r : U \times I \rightarrow R$, mapping each $(u, i) \in U \times I$ to rating set R . In other words, with the rating function r , it estimates values for unknown user-item pairs, \hat{r}_{ui} in the Y matrix [6].

2.2 Multi-Criteria Recommendation

Let U, I, D be the set of users, items, and criteria with the size of M, N , and K , respectively. Y is the dataset consisting of the known ratings r_{uid} which represents the known rating of user u to item i for the criteria d which respect to $u \in U$, $i \in I$, and $d \in D$. The rating values r_{uid} should be normalized, for example, Tripadvisor uses a Likert scale of 5 values $\{1, 2, 3, 4, 5\}$ for $\{Terrible, Poor, Average, Very good, Excellent\}$, respectively. The multi-criteria recommendation problem is to predict the \hat{r}_{uid} that is unknown for certain users, items, and criteria. There are usually two steps: (i) multi-criteria rating predictions, which is the process of predicting the rating on each criterion, and (ii) rating aggregations which refer to the process of aggregating the predicted multi-criteria ratings to estimate the overall rating. The recommended items will be produced based on these estimated overall ratings [4].

2.3 Context-Awareness

A simple definition of context that is widely used by many studies is presented in [7], according to which “context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves”. For example, considering a tourism recommender system, the entities are travelers and places. In particular, contextual factors can affect the user’s experience, such as his mood, companion, or weather condition. A context-aware travel recommender system must use this contextual information to recommend the most suitable tourism destinations for the user according to his situation.

According to Dourish [8], context can be classified into two views: the representational view and the interactional view. In the first view, context describes the circumstances in which a user chooses or experiences an item. It is represented through a known set of properties that do not change over time. For example, when a user listens to a track, the contextual factors that influence the user’s experience can be their *mood* and the *space* in which they listen to the music. The *mood* context factor can take values such as *happiness*, *sadness*, and *stress*. The *space* can take values such as *quiet*, *noisy*. A list of contextual factors and their value has been selected before in the process of building the system. At the same time, the interactional view assumes a cyclical relationship between context and activity, where the activity gives rise to context and the context influences activity [6]. Unlike the representational context, the user’s interactional context is hidden through a series of user interactions with the system. It expresses the intention and the long- and short-term preferences of the user. For example, a web server log records the user’s actions when they view a product, like it, add it to a favorites list, put it in a shopping cart, or pull it out of a shopping cart. These are important contextual information that helps the recommender system make appropriate recommendations.

2.4 Paradigms for Using Contextual Information

Traditional recommender systems are based on the 2-dimensional (2D) data set $Users \times Items$, also known as the utility matrix. A recommender function is built on this data set to predict user preferences for unrated items, thereby ranking the results and producing a list of the most relevant items for users. The traditional recommendation process consists of three components: input, recommendation function, and output, which can be presented as follows: $[Data (input)] \rightarrow [Recommendation function] \rightarrow [Predicted ratings/Ranking \rightarrow Recommendation items (output)]$.

Adomavicius et al. [3] have classified CARS into three paradigms depending on which component of the recommendation process the contextual information is included: contextual pre-filtering, contextual post-filtering, and contextual modeling.

- Contextual pre-filtering: this approach impacts the input component in the recommendation process. Specifically, contextual information is used to select or construct a dataset that is appropriate for the given context. Then, traditional 2D recommender systems will be applied to this new dataset.
- Contextual post-filtering: traditional 2D recommender systems are used directly on the input dataset without taking into account context, and then, the results are changed based on contextual information in the output component.
- Contextual modeling: the contextual information is integrated directly into the recommendation function.

2.5 Context-Aware Multi-Criteria Recommendation Problem Statement

Context-aware multi-criteria recommender systems (CA-MCRSs) are an extension of MCRSs, giving recommendations to users and accounting for contextual information (e.g., weather, time, user's mood) or latent contexts [6,9]. On the other hand, user preference data are expanded into a multidimensional dataset, including users, items, and contextual information.

Formally, denoting with C a set of contextual information, $c \in C$ is a context that represents the situation when the user experiences the item. The CA-MCRS problem with the rating function is defined as:

$$r : U \times I \times C \times D \rightarrow R \quad (1)$$

where \hat{r}_{uicd} is the predicted value of the rating function r for user u on item i with the criteria d in context c or for mapping each set $(u, i, c, d) \in U \times I \times C \times D$ into the set of rating values R .

3 The Proposed Method

Our goal is to recommend a subset of the unknown items to a given user. In the steps to perform this goal while using deep learning for CA-MCRS, there are three main tasks: (1) context-aware multi-criteria rating predictions, (2) rating aggregations, and (3) recommendation. The general architecture of our proposed method is shown in Fig. 1. To integrate contextual information, we apply item-splitting [10], and then use DNNs to predict the K context-aware multi-criteria ratings in the first step, and learn the aggregation function f in the second step, as illustrated in Fig. 2.

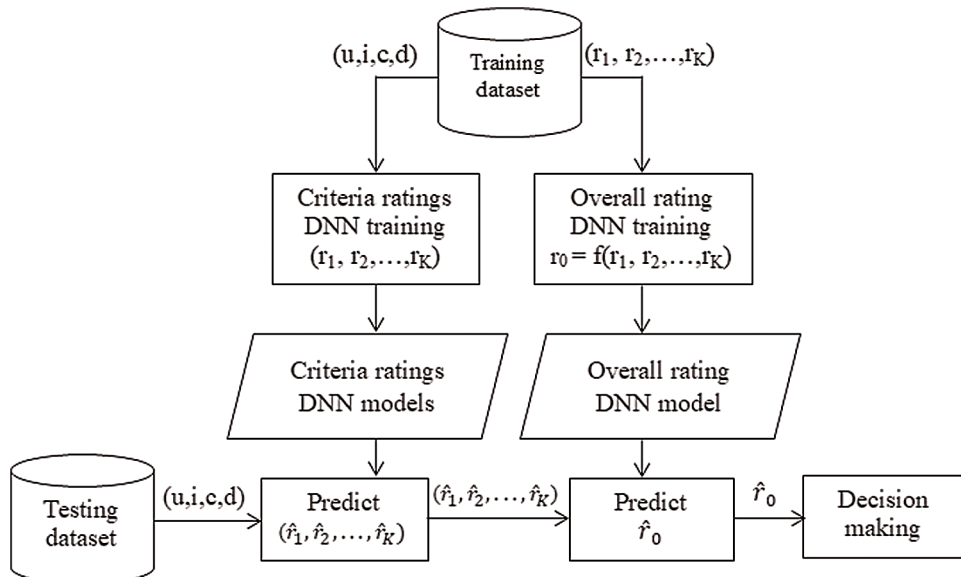


Figure 1: Overview of the context-aware multi-criteria recommender system

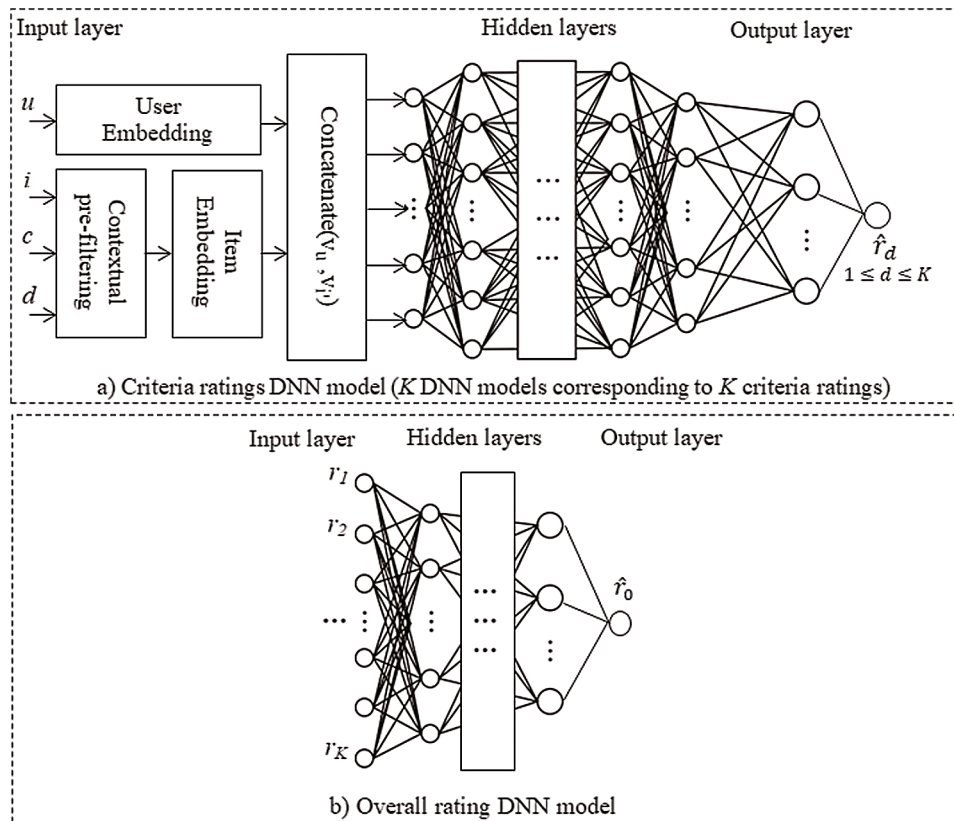


Figure 2: Architecture of DNN models: (a) Context-aware multi-criteria rating predictions and (b) Rating aggregations

3.1 Context-Aware Multi-Criteria Rating Predictions

In this step, we use DNNs to predict the criteria ratings for a user on an item in a specific context as Fig. 2a. To do this work, we train K DNN models corresponding to K criteria ratings. The process to predict the rating for each criterion is as follows:

First, we apply item-splitting into the dataset to remove the contextual dimension. Item-splitting is the first splitting-based algorithm in the contextual pre-filtering approach, which was proposed by Baltrunas et al. [10]. The main idea is that a rating on an item can be influenced by contextual factors, and thus, we can split the set of ratings for each item into two groups according to the target context (e.g., one group includes ratings in the context *Spring*, and the other includes ratings in remain seasons). This splitting is performed only on the item if it forms two groups with a statistically significant difference. After this stage, each item ID in the original dataset can be converted to two new item IDs if this item is split by a contextual condition. In other words, we obtain a new two-dimensional matrix $Users \times Items$, with the number of items that can be increased depending on the number of items split. So for each tuple user ID, item ID, and context ID (denoted by (u, i, c)), through the contextual pre-filtering process, they are converted to user ID and new item ID (denoted by (u, i') , with i' being the newly generated item ID). This tuple is fed to the DNN in the next step.

Second, our DNN model start with the input layer, we convert each user ID and new item ID into a dense continuous valued and low dimensional vector called an embedding vector. The embedding vectors are

initialized with random values, and then they are adjusted to minimize the loss function during the model training. The criteria ratings DNN model is shown in Fig. 2a. The input vector x is given by:

$$x = \text{Concatenate}(v_u, v_i) \quad (2)$$

where v_u and v_i are user and item embedding vectors.

Third, the DNN model continues with a series of hidden layers. We use the Rectified Linear Units (ReLU) as the activation function:

$$\text{ReLU}(z) = \max(0, z) \quad (3)$$

where z is the input of a hidden layer. The output of a hidden layer l is formulated as:

$$a^{(l)} = \text{ReLU}\left(W^{(l)}a^{(l-1)} + b^{(l)}\right) \quad (4)$$

where $W^{(l)}$ and $b^{(l)}$ are the weight matrix and bias vector for the layer l and $a^{(1)} = x$.

Finally, there is 1 neuron in the output layer corresponding to a criteria rating. We predict the user criteria rating \hat{r}_{uid} (denoted by \hat{r}_d for short, $1 \leq d \leq K$) as follows:

$$\hat{r}_d = \text{ReLU}\left(W^{(L)}a^{(L-1)} + b^{(L)}\right) \quad (5)$$

where L is the number of layers.

3.2 Rating Aggregations

In a conventional way, the overall rating is predicted based on a two-dimensional matrix $Users \times Items$. The algorithms in this way learn a rating function that captures the interaction between users and items. In MCRSs, the overall rating can be estimated by an aggregation function of multi-criteria ratings. In this step, we followed the same as in [11]. Specifically, we learn the relationship between the overall rating r_0 and the criteria ratings r_1, r_2, \dots, r_K .

$$r_0 = f(r_1, r_2, \dots, r_K) \quad (6)$$

The input vector is the criteria ratings r_1, r_2, \dots, r_K , which fed to the input layer of the DNN as shown in Fig. 2b. We normalize the input vector as follows:

$$z_d = \frac{r_d - m}{s} \quad (7)$$

where z_d is the normalized criteria rating r_d , $1 \leq d \leq K$, m is the mean of the training samples and s is the standard deviation of the training samples. The input vector becomes:

$$x = [z_1, z_2, \dots, z_k]^T \quad (8)$$

Then, in the hidden layers, ReLUs are used as the activation functions. The output of a hidden layer is given again by Eq. (4). In the output layer, we predict the overall rating r_0 as in Eq. (5).

3.3 Recommendation

The CA-MCRS makes recommendations for a user in a specified context based on the user's overall rating on the items predicted from trained DNN models. The recommendation process including three steps is shown in Fig. 1:

1. Predict criteria ratings: We predict the criteria ratings $\hat{r}_1, \hat{r}_2, \dots, \hat{r}_K$ using the criteria ratings DNNs as described in Section 3.1. The user, item, and context are fed to the models as inputs.
2. Predict overall ratings: We predict the overall rating \hat{r}_0 using the overall rating DNN as described in Section 3.2. The predicted criteria ratings $\hat{r}_1, \hat{r}_2, \dots, \hat{r}_K$ are normalized before feeding them as inputs to the model.
3. Provide recommendation: Finally, we recommend items with a high overall rating predicted in the previous step.

4 Experiments

4.1 Dataset

We used the public dataset, namely, TripAdvisor¹ [11]. This is a multi-criteria rating dataset, which contains users' ratings for hotels. It includes seven criteria ratings (Value, Rooms, Location, Cleanliness, Check-in/front desk, Service, and Business service) and an overall rating, the rating range between 1 and 5, details as in Tab. 1. Based on the timestamp, we create additional contextual dimensions such as year, month, and quarter. In experiments, we used quarters of the year as contextual information in the dataset. Note that it is difficult to collect datasets with both contextual and multi-criteria information.

Table 1: Datasets properties

Field description	Users	Items	Ratings	Sparsity
Value	6,136	1,840	244,006	97.84%

4.2 Metric Evaluation

The MAE and RMSE are common metrics for the evaluation rating prediction task [12], which are based on the error magnitude, which means the difference between the predicted rating and the actual rating. Let T be an evaluation dataset that consists of pairs (u, i, c) with hidden rating values. The MAE and RMSE measures are given in Eqs. (9) and (10), respectively.

$$MAE = \frac{1}{|T|} \sum_{(u,i,c) \in T} |\hat{r}_{uic} - r_{uic}| \quad (9)$$

$$RMSE = \sqrt{\frac{1}{|T|} \sum_{(u,i,c) \in T} (\hat{r}_{uic} - r_{uic})^2} \quad (10)$$

4.3 Experimental Setup

To evaluate the effectiveness of our method, we compare to benchmark methods from the relevant state-of-art. More specifically, we design two categories as baselines to evaluate the effectiveness of (i) criteria rating prediction and (ii) overall rating prediction, as follows:

1. Deep multi-criteria collaborative filtering (DMCCF) model [11] - A benchmark for MCRS that employs deep learning and multi-criteria in recommendation systems.
2. Neural collaborative filtering (NCF) [13] - A benchmark for deep learning based recommender systems.

¹ <https://www.tripadvisor.com>

3. Context-aware neural collaborative filtering (CA-NCF) [14] - A benchmark for CARS that incorporates contextual information into deep learning based recommender systems.

To implement these models, we used Keras² with TensorFlow³ as the backend. We randomly initialized the DNN parameters using a normal distribution with a mean of 0 and a standard deviation of 0.05. We used Adam optimizer, with learning rates are set ranging from 0.01 to 0.0001, the epoch is set to maximum 20. We applied 5-fold cross-validation to produce the evaluation results.

In the criteria ratings DNN settings, we followed the same as in [11] to select the $[128 \rightarrow 64 \rightarrow 32 \rightarrow 16 \rightarrow 8]$ hidden layers, whereas, in the output layer, there is 1 neuron for 1 criteria rating. The item-splitting algorithm is used for contextual pre-filtering. There are 7 DNN models trained corresponding to 7 criteria ratings. In the overall rating DNN settings, we select the $[64 \rightarrow 32 \rightarrow 16 \rightarrow 8]$ hidden layers, whereas, in the output layer, there is 1 neuron for the overall rating.

We compare the overall rating prediction performance on two aspects: (i) between using contextual information and not considering contextual information; (ii) between using traditional dimensions in the utility matrix (i.e., users, items, and contexts) and inferring from the criteria ratings. Specifically, there are four overall rating prediction models used in our experiment as follows:

1. Overall rating prediction based on two-dimensional matrix Users \times Items: In this case, the contextual information is ignored while user ID and item ID pairs are fed to the DNN model as inputs [13].
2. Overall rating prediction based on multidimensional matrix Users \times Items \times Contexts as in [14]: The item-splitting algorithm is used to transform the multidimensional matrix to the two-dimensional matrix, and then use DNNs to predict the overall ratings based on the pairs of user IDs and new item IDs.
3. Overall rating prediction based on criteria ratings predicted in the previous step by using only DNN (ignore contextual information) as in [11]: Seven criteria ratings are normalized and then fed to the DNN model as inputs.
4. Overall rating prediction based on criteria ratings predicted in the previous step by using contextual pre-filtering + DNN (CA-MCRS): Seven criteria ratings are normalized and then fed to the DNN model as inputs.

4.4 Results and Discussion

In the contextual pre-filtering stage, for each criteria rating (i.e., $r_1 \dots r_7$), we split the items into two virtual items based on contextual conditions. Tab. 2 shows some cases where items are split according to the corresponding contextual condition. The higher the *Statistic* value indicates the presence of a contextual condition that makes a significant difference in the user's ratings. The threshold *p-value* is chosen to be less than 0.05, the lower the *p-value*, the greater the statistical significance of the observed difference. As a result of the contextual pre-filtering stage, the number of items in the dataset increases from the original depending on how many items are split.

In Figs. 3 and 4, we portray the performances of criteria rating prediction in terms of the MAE and RMSE for two methods: (i) the combination of contextual pre-filtering and DNN model, and (ii) using only the DNN model. We can see that the combination of contextual pre-filtering and DNN gives better results than using only the DNN model on all criteria ratings. The average MAE and RMSE across 7 criteria ratings of the DNN model are 0.8397 and 1.0877, meanwhile, that of the DNN model with integrated contextual pre-filtering are 0.8179 and 1.0636 respectively.

² <https://github.com/keras-team/keras>

³ <https://github.com/tensorflow/tensorflow>

Table 2: Examples of some cases where items are split by contextual conditions

Item ID	Contextual condition	Contextual factor	Statistic	p-value
81782	Q2	Quarter	4.0842	0.0001
189389	Q3	Quarter	3.8209	0.0002
197792	Q2	Quarter	3.7822	0.0004
218479	Q1	Quarter	3.4578	0.0006
100561	Q4	Quarter	3.3947	0.0010
257798	Q2	Quarter	3.3727	0.0023
302020	Q1	Quarter	3.3580	0.0010
85003	Q3	Quarter	3.3493	0.0014

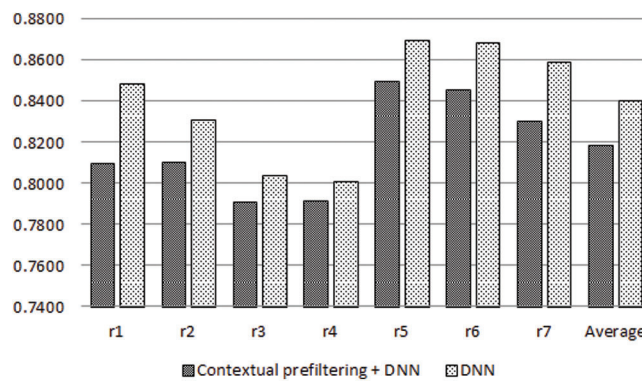
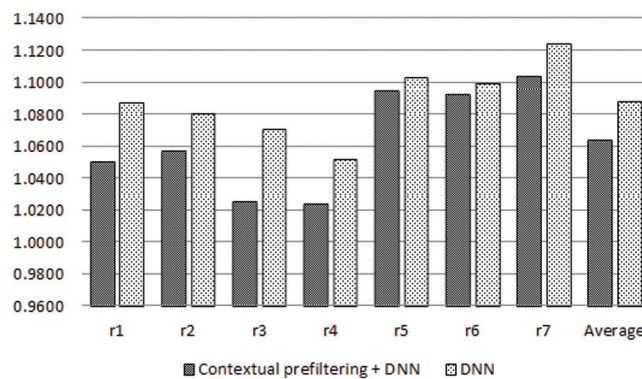
**Figure 3:** Results of criteria rating prediction on MAE metric**Figure 4:** Results of criteria rating prediction on RMSE metric

Fig. 5 presents the experimental results of overall rating prediction in terms of the MAE and RMSE for all four methods, including NCF [13], CA-NCF, [14], DMCCF [11], and CA-MCRS (our method). The results demonstrate that our method provides better results than all remaining methods. The results also show that,

- The overall rating prediction based on the criteria ratings gave better results than the traditional way based on the two-dimensional matrix $Users \times Items$. Specifically, the DMCCF method gives a smaller

error, MAE = 0.7995, RMSE = 1.0988, compared to the NCF method with MAE = 0.8215, RMSE = 1.1327

- By integrating contextual information, the models generate more accurate predictions. Specifically, the CA-NCF method (with MAE = 0.8102 and RMSE = 1.1078) gives better results than the NCF method (with MAE = 0.8215 and RMSE = 1.1327), and our method, CA-MCRS (with MAE = 0.7792 and RMSE = 1.0521), gives better predictions than DMCCF (with MAE = 0.7995 and RMSE = 1.0988)

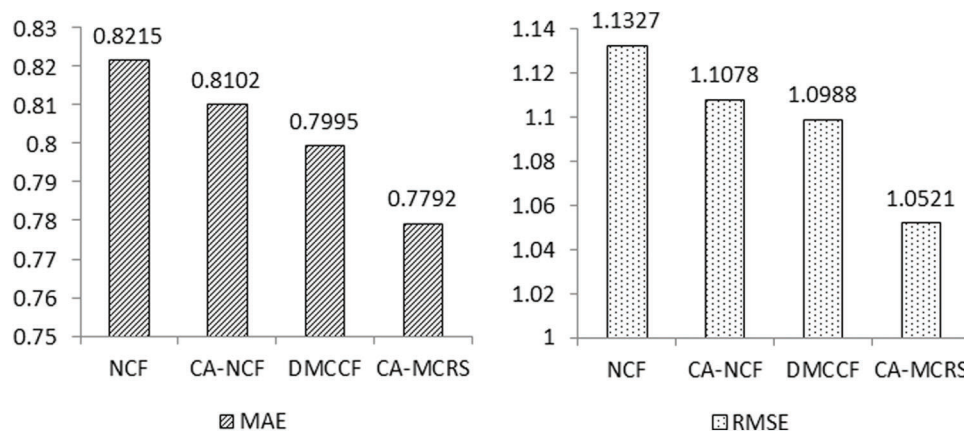


Figure 5: Results of overall rating prediction on MAE and RMSE metrics

All the aforementioned experimental results show that integrating context makes predictive models more efficient for both criteria ratings and overall ratings. Empirical evidence also shows that predicting the overall rating based on aggregating criteria ratings gives better results than using the traditional dimensions on the utility matrix. Our method combines contextual information and criteria ratings to predict overall rating, thereby making more relevant recommendations.

5 Related Works

The integration of contextual information into recommender systems to improve their performance has recently received considerable attention in the research literature [15–21]. Studies in the relevant literature can be classified into three paradigms depending on which component of the recommendation process the contextual information is included in: contextual pre-filtering, contextual post-filtering, and contextual modeling. The contextual pre-filtering paradigm impacts the input component in the recommendation process. Specifically, contextual information is used to select or construct a dataset that is appropriate for the given context. Then, traditional 2D recommender systems will be applied to this new dataset. Some techniques are used in this paradigm such as reduction-based [22,23] and splitting-based [10,24]. With the contextual post-filtering paradigm, traditional 2D recommender systems are used directly on the input dataset without taking into account context. The contextual information is then used to adjust the outputs in two ways: filtering out or adjusting the result [3]. Some contextual post-filtering methods are Weight PoF and Filter PoF [25,26]. An alternative paradigm that is integrated directly into the recommendation function. This paradigm can be divided into two classes: neighborhood-based and model-based. Tensor Factorization [27], CAMF [28], and CSLIM [29] are model-based methods with integrated contextual information. Differential context modeling [30,31] is an example of the neighborhood-based method.

Multi-criteria decision support systems help to make decisions based on different criteria. In fact, decisions are often built on analysis and synthesis on multiple criteria [32–35]. In recommender systems,

item attributes can be used as criteria (e.g., actors, story, and visual effects of a movie) when users give ratings. Making recommendations to users can exploit rating information about these aspects. MCRS utilizes user preferences in multiple criteria to better generate recommendations. Several attempts have been made to handle MCRS. One of the popular techniques is memory-based which uses the item-based or user-based collaborative filtering methods to predict the sub-criteria scores, the overall rating is then obtained by an aggregation function. Another one is the model-based technique which constructs a predictive model to predict unknown ratings, such as Probabilistic Modeling [36], Support Vector Regression [37], and Multilinear Singular Value Decomposition [38]. Recently, Nassar et al. [11] employed deep learning and multi-criteria in the recommendation system.

Some works [4,5] have exploited the methods to combine CARS and MCRS. Zheng et al. [4] integrated contextual information into four baseline approaches for multi-criteria recommendations in educational learning. The independent and dependent methods were used for the multi-criteria rating predictions, and the linear and conditional aggregations for the rating aggregations. Dridi et al. [5] presented the spectral graph partitioning method, which consists in jointly clustering users involved in contextual situations while rating items with respect to multiple facets.

6 Conclusions and Future Work

Applying deep learning techniques to recommender systems is a modern approach for providing better recommendations. However, how to exploit context in deep learning-based recommender systems is still an open issue. In this paper, we proposed a novel approach, which relies on deep learning for CA-MCRSs. We showed how to integrate contextual information into MCRSs using deep learning. Specifically, we applied DNN models to predict the context-aware multi-criteria ratings and learn the aggregation function. Experiments have been conducted to evaluate the effect of this approach on the real-world dataset. A significant result is that our method outperforms other state-of-the-art methods for recommendation effectiveness.

It is worth to emphasize that this is the first attempt for incorporating contextual information into MCRSs using deep learning. As a future perspective, we will extend our work by incorporating contextual information into various deep learning techniques. Furthermore, we can use a single DNN model with multiple outputs trained in multi-label classification styles. Additionally, it would be interesting to investigate other activation functions on output nodes along with normalization methods for desired predicted value.

Funding Statement: This work is supported by project No. B2020-DQN-08 from the Ministry of Education and Training of Vietnam.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] M. Deshpande and G. Karypis, "Item-based top-n recommendation algorithms," *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 1, pp. 143–177, 2004.
- [2] Q. H. Le, S. L. Vu, T. K. P. Nguyen and T. X. Le, "A state-of-the-art survey on context-aware recommender systems and applications," *International Journal of Knowledge and Systems Science (IJKSS)*, vol. 12, no. 3, pp. 1–20, 2021.
- [3] G. Adomavicius and A. Tuzhilin, *Context-aware recommender systems*, in *Recommender systems handbook*. Boston, MA, USA: Springer, pp. 217–253, 2011.
- [4] Y. Zheng, S. Shekhar, A. A. Jose and S. K. Rai, "Integrating context-awareness and multi-criteria decision making in educational learning," in *Proc. of the 34th ACM/SIGAPP Sym. on Applied Computing*, Limassol, Cyprus, pp. 2453–2460, 2019.

- [5] R. Dridi, L. Tamine and Y. Slimani, "Context-aware multi-criteria recommendation based on spectral graph partitioning," in *Int. Conf. on Database and Expert Systems Applications*, Linz, Austria, pp. 211–221, 2019.
- [6] G. Adomavicius, B. Mobasher, F. Ricci and A. Tuzhilin, "Context-aware recommender systems," *AI Magazine*, vol. 32, no. 3, pp. 67–80, 2011.
- [7] A. K. Dey, "Understanding and using context," *Personal and Ubiquitous Computing*, vol. 5, no. 1, pp. 4–7, 2001.
- [8] P. Dourish, "What we talk about when we talk about context," *Personal and Ubiquitous Computing*, vol. 8, no. 1, pp. 19–30, 2004.
- [9] M. Unger, A. Bar, B. Shapira and L. Rokach, "Towards latent context-aware recommendation systems," *Knowledge-Based Systems*, vol. 104, no. 1-2, pp. 165–178, 2016.
- [10] L. Baltrunas and F. Ricci, "Context-based splitting of item ratings in collaborative filtering," in *Proc. of the third ACM conf. on Recommender systems*, New York, NY, USA, pp. 245–248, 2009.
- [11] N. Nassar, A. Jafar and Y. Rahhal, "A novel deep multi-criteria collaborative filtering model for recommendation system," *Knowledge-Based Systems*, vol. 187, no. 1, pp. 104811, 2020.
- [12] G. Shani and A. Gunawardana, *Evaluating recommendation systems*, in *Recommender systems handbook*. Boston, MA, USA: Springer, pp. 257–297, 2011.
- [13] X. He, L. Liao, H. Zhang, L. Nie, X. Hu *et al.*, "Neural collaborative filtering," in *Proc. of the 26th Int. Conf. on World Wide Web*, Perth, Australia, pp. 173–182, 2017.
- [14] M. Al Jawarneh, P. Bellavista, A. Corradi, L. Foschini, R. Montanari *et al.*, "A pre-filtering approach for incorporating contextual information into deep learning based recommender systems," *IEEE Access*, vol. 8, pp. 40485–40498, 2020.
- [15] L. Wu, C. Quan, C. Li, Q. Wang, B. Zheng *et al.*, "A context-aware user-item representation learning for item recommendation," *ACM Transactions on Information Systems*, vol. 37, no. 2, pp. 1–29, 2019.
- [16] S. Raza and C. Ding, "Progress in context-aware recommender systems - an overview," *Computer Science Review*, vol. 31, no. 1, pp. 84–97, 2019.
- [17] N. M. Villegas, C. Sánchez, J. Díaz-Cely and G. Tamura, "Characterizing context-aware recommender systems: A systematic literature review," *Knowledge-Based Systems*, vol. 140, no. 1, pp. 173–200, 2018.
- [18] G. Adomavicius, K. Bauman, B. Mobasher, F. Ricci, A. Tuzhilin *et al.*, "Workshop on context-aware recommender systems," in *Fourteenth ACM Conf. on Recommender Systems*, Virtual Event, Brazil, pp. 635–637, 2020.
- [19] S. Kulkarni and S. F. Rodd, "Context aware recommendation systems: A review of the state of the art techniques," *Computer Science Review*, vol. 37, no. 2, pp. 100255, 2020.
- [20] H. Liu, Y. Tong, P. Zhang, X. Lu, J. Duan *et al.*, "Hydra: A personalized and context-aware multi-modal transportation recommendation system," in *Proc. of the 25th ACM SIGKDD Int. Conf. on Knowledge Discovery & Data Mining*, Anchorage, AK, USA, pp. 2314–2324, 2019.
- [21] H. F. Abdulkareem, G. Y. Abozaid and M. I. Soliman, "Context-aware recommender system frameworks, techniques, and applications: A survey," in *2019 Int. Conf. on Innovative Trends in Computer Engineering (ITCE)*, Aswan, Egypt, pp. 180–185, 2019.
- [22] G. Adomavicius, R. Sankaranarayanan, S. Sen and A. Tuzhilin, "Incorporating contextual information in recommender systems using a multidimensional approach," *ACM Transactions on Information Systems*, vol. 23, no. 1, pp. 103–145, 2005.
- [23] V. Codina, F. Ricci and L. Ceccaroni, "Distributional semantic pre-filtering in context-aware recommender systems," *User Modeling and User-Adapted Interaction*, vol. 26, no. 1, pp. 1–32, 2016.
- [24] Y. Zheng, B. Mobasher and R. D. Burke, "The role of emotions in context-aware recommendation," in *Proc. of the 2013 Decisions@RecSys Workshop in conjunction with the 7th ACM Conf. on Recommender Systems*, Hong Kong, China, pp. 21–28, 2013.
- [25] U. Panniello, A. Tuzhilin, M. Gorgoglione, C. Palmisano and A. Pedone, "Experimental comparison of pre-vs. post-filtering approaches in context-aware recommender systems," in *Proc. of the third ACM Conf. on Recommender Systems*, New York, NY, USA, pp. 265–268, 2009.

- [26] X. Ramirez-Garcia and M. García-Valdez, "Post-filtering for a restaurant context-aware recommender system," in *Recent Advances on Hybrid Approaches for Designing Intelligent Systems*, Springer, pp. 695–707, 2014.
- [27] A. Karatzoglou, X. Amatriain, L. Baltrunas and N. Oliver, "Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering," in *Proc. of the fourth ACM Conf. on Recommender Systems*, Barcelona, Spain, pp. 79–86, 2010.
- [28] L. Baltrunas, B. Ludwig and F. Ricci, "Matrix factorization techniques for context aware recommendation," in *Proc. of the fifth ACM Conf. on Recommender Systems*, Chicago, Illinois, USA, pp. 301–304, 2011.
- [29] Y. Zheng, B. Mobasher and R. Burke, "Deviation-based contextual slim recommenders," in *Proc. of the 23rd ACM Int. Conf. on Conf. on Information and Knowledge Management*, Shanghai, China, pp. 271–280, 2014.
- [30] Y. Zheng, R. Burke and B. Mobasher, "Differential context relaxation for context-aware travel recommendation," in *Int. Conf. on Electronic Commerce and Web Technologies*, Vienna, Austria, pp. 88–99, 2012.
- [31] Y. Zheng, R. Burke and B. Mobasher, "Recommendation with differential context weighting," in *Int. Conf. on User Modeling, Adaptation, and Personalization*, Rome, Italy, pp. 152–164, 2013.
- [32] F. Hdioud, B. Frikh and B. Ouhbi, "Multi-criteria recommender systems based on multi-attribute decision making," in *Proc. of Int. Conf. on Information Integration and Web-based Applications & Services*, Vienna, Austria, pp. 203–210, 2013.
- [33] C. N. Wang, T. T. Dang and N. A. T. Nguyen, "Location optimization of wind plants using DEA and fuzzy multi-criteria decision making: A case study in Vietnam," *IEEE Access*, vol. 9, pp. 116265–116285, 2021.
- [34] N. B. T. Nguyen, G. H. Lin and T. T. Dang, "Fuzzy multi-criteria decision-making approach for online food delivery (OFD) companies evaluation and selection: A case study in Vietnam," *Processes*, vol. 9, no. 8, pp. 1274, 2021.
- [35] C. N. Wang, T. T. Dang, N. A. T. Nguyen and J. W. Wang, "A combined Data Envelopment Analysis (DEA) and Grey Based Multiple Criteria Decision Making (G-MCDM) for solar PV power plants site selection: A case study in Vietnam," *Energy Reports*, vol. 8, no. 4, pp. 1124–1142, 2022.
- [36] N. Sahoo, R. Krishnan, G. Duncan and J. Callan, "Research note—the halo effect in multicomponent ratings and its implications for recommender systems: The case of yahoo! movies," *Information Systems Research*, vol. 23, no. 1, pp. 231–246, 2012.
- [37] H. Drucker, C. J. Burges, L. Kaufman, A. Smola and V. Vapnik, "Support vector regression machines," in *Proc. of the 9th Int. Conf. on Neural Information Processing Systems*, Denver, Colorado, USA, pp. 155–161, 1996.
- [38] M. Hassan and M. Hamada, "A neural networks approach for improving the accuracy of multi-criteria recommender systems," *Applied Sciences*, vol. 7, no. 9, pp. 868, 2017.