Context-Aware Multi-Criteria Recommender Systems Using Variable Selection Networks

Ngoc Luyen Le, Marie-Hélène Abel

Context-Aware Multi-Criteria Recommender Systems Using Variable Selection Networks

LE Ngoc Luyen, Marie-Hélène ABEL

Université de technologie de Compiègne, CNRS, Heudiasyc (Heuristics and Diagnosis of Complex Systems), CS 60319 - 60203 Compiègne Cedex, France

Abstract. Conventional recommender systems, which rely on a single criterion such as overall rating, often fail to capture the complexity of user preferences and the influence of contextual information. Context-aware multi-criteria recommender systems address these limitations by incorporating multiple dimensions of user preferences, item attributes, and contextual factors, leading to more accurate and relevant recommendations. This paper presents a context-aware multi-criteria recommender system using variable selection networks. Our approach dynamically selects the most relevant features from a variety of inputs, including user profiles, item characteristics, multiple criteria, and contextual factors, to enhance personalization. By leveraging deep learning-based variable selection networks, our model significantly improves recommendation accuracy and interpretability, outperforming several baseline models in experimental evaluations. This advancement underscores the importance of integrating both multi-criteria and context-aware methodologies in modern recommender systems.

Keywords: Recommender System · Feature Selection · Context-aware Recommender System · Deep Learning.

1 Introduction

Recommender systems have become an integral part of many platforms, providing personalized recommendations to users across various domains such as e-commerce, entertainment, education, or social media [20], [11], [3], [12]. Traditional recommender systems typically rely on a single criterion, such as overall rating or purchase history, to make suggestions. While these systems have been successful in enhancing user experience and driving engagement, they often fail to capture the multifaceted nature of user preferences. Users usually consider multiple factors when making decisions depending on specific features of a product or service.

Multi-criteria Recommender Systems (MCRS) address this limitation by incorporating multiple criteria into the recommendation process. By considering various dimensions of user preferences and item attributes, these systems can provide more accurate and satisfying recommendations [9]. For instance, in the context of e-learning platforms, learners may evaluate courses based on different criteria such as content relevance, instructor effectiveness, course difficulty,

and interactive elements. A MCRS can analyze these diverse evaluations to suggest courses that align more closely with the learner's specific educational goals and preferences. Considering contextual information in MCRSs further enhances their ability to deliver personalized and relevant recommendations [29]. Contextual factors such as the learner's current knowledge level, time of access, device used, and even their learning environment can significantly influence their preferences and needs. For example, a learner accessing an e-learning platform on a mobile device during a short break might prefer quick, concise lessons, while the same learner at home on a desktop may favor more in-depth, comprehensive content.

By integrating contextual information and multiple criteria, advanced recommender systems can more effectively understand and model user preferences, resulting in more personalized and relevant recommendations. The evolution of deep learning has significantly improved the accuracy of recommendation tasks, with various models contributing to these advancements. Among these, Variable Selection Networks (VSNs) stand out as a powerful deep learning architecture designed to dynamically select and prioritize the most relevant features from a diverse set of inputs [17]. VSNs are particularly effective in managing the complexity of multiple criteria and contextual factors, making them highly suitable for recommendation tasks.

In this paper, we present the development of a Context-Aware Multi-Criteria Recommender System using Variable Selection Networks. Our approach leverages the strengths of VSNs to significantly enhance the personalization and relevance of recommendations by dynamically adapting to the most critical features, which are identified based on user preferences, item characteristics, and contextual factors. To demonstrate the effectiveness of our approach, we conducted experiments on an educational dataset. The results show that our approach outperforms several baseline models across different evaluation metrics. This success highlights the potential of using VSNs in context-aware MCRSs, providing more precise and adaptable solutions across various domains.

The remainder of this paper is organized as follows: In section 2, we review related work on context-aware MCRSs. Following this, Section 3 outlines our primary contributions, including problem formulation and the architcture of a context-aware MCRS based on VSN. In section 4, we present experimental evaluations of our approach. Finally, we conclude the paper in the last section.

2 Related Work

The development of recommender systems has evolved significantly, with initial methods focusing on single-criterion models. However, the need to capture multi-dimensional user preferences has led to the emergence of MCRS [13], [15]. This section reviews methodologies in the field of MCRS, highlighting advancements in leveraging deep learning-based approaches and contextual information.

As shown in the table 1,we identify two principal approaches for MCRSs: Multi-Criteria Collaborative Filtering, deep neural networks-based. In general,

Adomavicius and Kwon [1] were pioneers in the field of MCRSs, extending traditional collaborative filtering to incorporate multiple dimensions of user preferences. Their work demonstrated that accounting for diverse user preferences significantly enhances recommendation accuracy compared to single-criterion approaches. Matrix factorization techniques in MCRS have been instrumental in uncovering latent factors that drive complex user-item interactions by decomposing the user-item rating matrix into lower-dimensional latent spaces. This approach, exemplified by Koren and colleagues [10], enables precise predictions of user preferences across various criteria. Additionally, hybrid methodologies combining collaborative and content-based filtering, as explored by Manouselis and colleagues [18], effectively address the cold-start problem and enhance recommendation quality by integrating user profiles and item attributes.

Table 1. Summary of Approaches in Context-Aware Multi-Criteria Recommender Systems

Approach	**Description**	**Advantage**	**Drawback**	**Works**
Multi-criteria Collaborative Filtering	Utilized matrix factorization to uncover latent factors driving user-item interactions in MCRS.	Precise predictions of user preferences across multiple criteria through latent space representation.	Limited scalability and complexity in handling multiple criteria.	[1] [10], [18]
Deep neural networks	Introduced deep neural networks in collaborative filtering for modeling complex, non-linear interactions in MCRS.	Better modeling of user-item interactions, leading to improved recommendation accuracy.	High computational cost and potential overfitting with deep networks.	[7], [4], [19], [21], [26], [2]

Deep neural networks has significantly transformed recommender systems by modeling complex, non-linear interactions between users and items [27], [6], [16], [23], [14]. Neural Collaborative Filtering (NCF), introduced by He and colleagues [7], was the first to apply deep neural networks in this context. Building on NCF, methods like AE-MCCF use autoencoders to address data sparsity in multi-criteria settings, improving accuracy [4]. Stacked Autoencoders further refine user preferences but face challenges in interpretability and efficiency [24]. Nassar and colleagues [19] developed a deep learning-based MCRS that models user and item features simultaneously, capturing complex multi-criteria relationships. Shambour and colleagues [21] advanced this by using deep learning to identify intricate data patterns, enhancing recommendation precision. Wang and colleagues [26] proposed a hybrid model combining deep learning with tensor factorization to capture complex user preferences, though this approach is computationally demanding and relies heavily on data quality.

Recent research in MCRS has increasingly integrated context to improve recommendation accuracy and relevance. Afzal and colleagues [2] developed a unified neural network to handle multiple contexts and criteria simultaneously, predicting context-aware multi-criteria ratings and synthesizing them into overall recommendations. Vu and colleagues [25] focused on enhancing contextual features, dynamically adjusting criteria weights, and clustering users based on context. Dridi and colleagues [5] proposed a hybrid approach combining collaborative, content-based, and contextual data, using high-order co-clustering to predict multi-criteria ratings by grouping users, contexts, and criteria based on their inherent dependencies.

Integrating contextual information allows MCRS to deliver more personalized and situationally appropriate recommendations. The use of deep learning-based approaches enables a more nuanced understanding of complex user behaviors, which can lead to improved predictive accuracy and finer segmentation of user preferences. In this paper, we will focus on these two aspects to enhance the generation of item predictions for users. In the next section, we will delve into our approach to developing an MCRS.

3 Context-Aware Multi-Criteria Recommender Systems: Our Approach

In this section, we first formalize the problem of designing a MCRS that incorporates both multiple criteria and contextual information to predict the overall rating for non-interacted items. Then, we present our deep neural network architecture for carrying out this task.

3.1 Problem Formulation

Given that $U = \{u_1, u_2, ..., u_n\}$ represents the set of users, $I = \{i_1, i_2, ..., i_m\}$ represents the set of items, $C = \{c_1, c_2, ..., c_p\}$ represents the set of contexts affecting user interactions with items, $CR = \{cr_1, cr_2, ..., cr_o\}$ represents the set of criteria through ratings, and $R = \{r_1, r_2, ..., r_q\}$ represents the set of overall ratings, the function of the MCRS can be defined as follows:

$$f(r) : U \times I \times C \times CR \longrightarrow R \tag{1}$$

where $U \times I \times C \times CR$ represents the Cartesian product of users, items, contexts, and criteria, covering all possible user-item interactions influenced by different contextual factors and criteria. CR encompasses multiple criteria ratings, providing an evaluation of items depending on various aspects. Each criteria rating cr is given by user u to item i under context c. The overall rating R is derived from the evaluation of the user for an item on a given context. This overall rating value serves as the primary metric for generating recommendations in the MCRS. For example, as shown in table 2, students have evaluated various projects based on multiple contexts and criteria. The contexts (C) include Class (c_1), Semester

(c_2), and Lockdown (c_3). The criteria (CR) for assessing the projects are Applicability (cr_1), Data Quality (cr_2), and Ease of Use(cr_3). Each entry in the table contains the student id (U), the project (I), the corresponding contexts and criteria ratings, and the overall rating (R).

Table 2. Student evaluate projects under various contexts and criteria

U	I	C			CR			R
		c_1	c_2	c_3	cr_1	cr_2	cr_3	
1140	Credit Card Fraud Detection	DM	Spring	POS	5	5	5	5
1287	Payroll Information System	DA	Fall	PRE	4	4	4	2
1335	News Website	DA	Spring	PRE	4	5	5	3
1481	Social Networks	DM	Spring	POS	2	3	4	3

Given the user-item-context data with multi-criteria ratings and a list of non-interacted items I_{ni} for each user, the main problem is to accurately predict a list of top-K items I_{topK} from I_{ni} for each user u. These items are anticipated to receive the highest overall ratings based on existing multi-criteria ratings and historical data. Therefore, the objective of the MCRS is to recommend the top-K non-interacted items I_{topK} to each user u by leveraging multi-criteria rating and contextual information, including an overall rating for each item. The system aims to maximize the relevance and satisfaction of these recommendations by focusing on items that users have not yet interacted with, but are predicted to rate highly based on different criteria rating and contextual information.

In the context of student and projects mentioned above, suppose we aim to recommend projects to students based on their past interactions and ratings. For instance, consider a student who has not yet interacted with the "News Website" and "Social Networks" projects. By analyzing the multi-criteria ratings and contextual information from similar students, the context-aware MCRS can predict the overall ratings that this student might assign to these projects. If the predicted ratings for "News Website" and "Social Networks" are among the highest, these projects would be included in the top-K recommendations I_{topK} for the student. Therefore, this personalized recommendation ensures that the student is presented with projects that align closely with their preferences and the contextual factors influencing their choices. In the following section, we will delve into our deep neural network architecture designed to effectively perform such recommendations using structured data.

3.2 Our Deep Neural Network Architecture Leveraging Variable Selection Networks

We delve into an architecture designed to enhance the performance of MCRS by employing dynamic feature selection based on contextual information. The model leverages VSNs to adaptively determine the most relevant features from

a diverse set of inputs, including user, item, context, and various criteria. By focusing on the most pertinent information, the model excels at processing and integrating complex data, dynamically weighting each feature's importance based on the specific context and criteria at hand. This approach ensures that the recommendations generated are both precise and contextually relevant, addressing the nuanced needs of users more effectively than traditional methods.

As illustrated in Figure 1, the general architecture is composed of several key components and layers, each playing an important role in enhancing overall performance. Below, we introduce each component and layer with details of their functionality.

Input and Embedding Layer: The input and embedding layer are designed to capture and transform diverse input data into dense vector representations, which are crucial for effective processing in subsequent layers of the model. The input layer receives various types of input data $x = \{U, I, C, CR\}$ from the set of user U, the set of item I, the set of context C and the set of criteria CR.

Each input passes through an embedding layer, which transforms the input identifiers into dense vectors. This transformation is essential for capturing latent relationships among the entities. Each embedding layer includes L2 regularization to prevent overfitting. The embedding transformation for each input x is given by:

$$e(x) = W_e(x) \tag{2}$$

where W_e is the embedding matrix for the input x. For different inputs, the embeddings correponding user embedding: $e(U) = W_U U$, item embedding $e(I) = W_I I$, context embedding $e(C) = W_c C$, and criteria embedding $e(CR) = W_{CR} CR$.

Concatenate and Full Connected Layer 1: Following the input and embedding layers, the architecture proceeds with a concatenation step and then processes the combined features through a fully connected (dense) layer. These layers are essential for integrating and refining the features extracted from the inputs. After the embedding layers transform the input identifiers into dense vector representations, these vectors are directly concatenated. The next step is to concatenate all these embeddings into a single, unified representation vector z_{dl1} as follows:

$$z_{dl1} = Concatenate(e(U), e(I), e(C), e(CR)) \tag{3}$$

This concatenated vector z_{dl1} contains the integrated information from all the input features. This vector is then passed through the first fully connected layer (i.e dense layer). In our architecture, this dense layer applies a linear transformation followed by a non-linear activation function of Rectified Linear Unit – *ReLU* – enabling the model to learn complex pattern in the data. The output of the first fully connected layer h_1 is given by:

$$h_1 = ReLU(W_1 z_{dl1} + b_1) \tag{4}$$

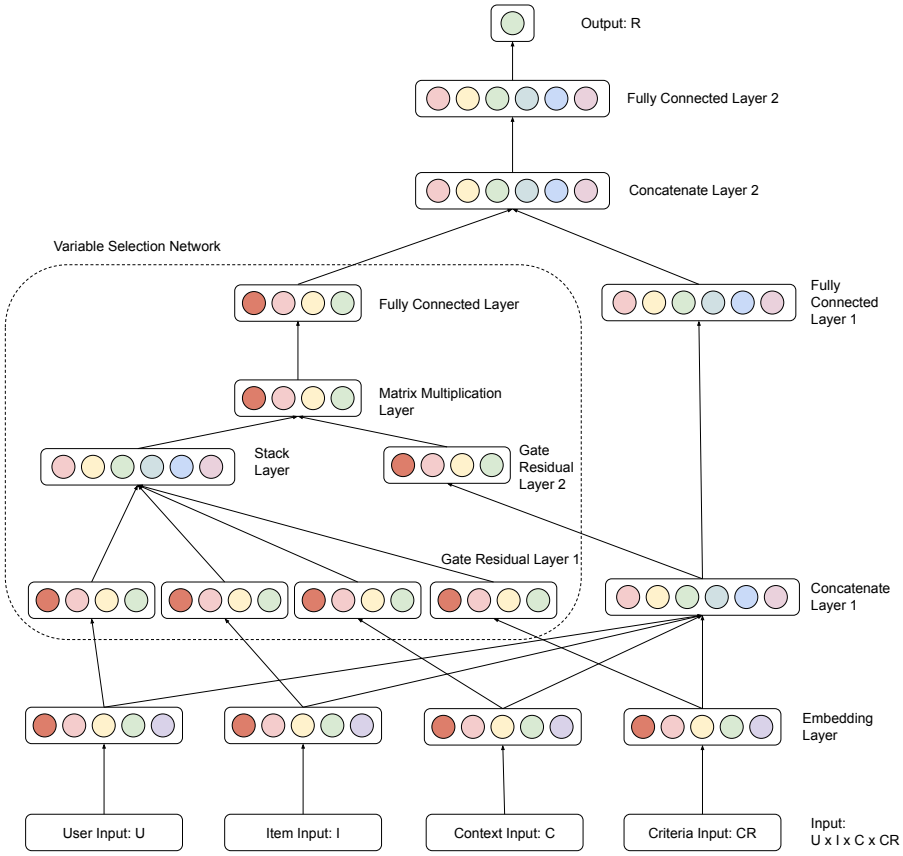

Fig. 1. The deep neural network architecture for the MCRS using Gated Residual and Variable Selection Networks.

where W_1 is the weight matrix for the first fully connected layer. b_1 is the bias vector for the first fully connected layer. $ReLU$ is the activation function defined as $ReLU(x) = max(0, x)$.

Variable Selection Network Component: Following the input and embedding layers, the VSN plays an essential role in the new architecture by dynamically selecting and emphasizing the most relevant features from the input embeddings. This selection mechanism ensures that the model prioritizes the most important information, enhancing both the accuracy and contextual relevance of the recommendations. The VSN operates as a separate branch in parallel with the precedent layer of the architecture and consists of several key elements:

Each input feature embedding is processed through its dedicated own Gated Residual Layer (GRL) to capture complex interactions and relationships independently. Let $z_U = e(U)$, $z_I = e(I)$, $z_C = e(C)$, and $z_{CR} = e(CR)$ be the

embedding for the user, item, context and criteria, respectively. Through the first GRL layer, each embedding z_i then is processed as follows:

$$g_i = GRL_i(z_i) = W_{g1}z_i + b_{g1} + \sigma(W_{g2}(W_{g1}z_i + b_{g_1})) \tag{5}$$

where W_{g1} and W_{g2} are weight matrices for the GRL, b_{g1} is the bias vector, and σ is the sigmoid activation function.

These GRL outputs for each embedding is stacked along a new axis to form 3D tensor G as follows:

$$G = Stack([g_U, g_I, g_C, g_{CR}]) \tag{6}$$

In parallel, the second GRL processes all input feature embeddings in the form of a single vector, benefiting from the output z_{dl1} of the first concatenation layer. This GRL captures complex interactions between the combined features and produces a transformed output that reflects these relationships as follows:

$$g_{cc} = GRL_{concat}(z_{dl1}) = W_{g_{cc1}}z_{dl1} + b_{g_{cc1}} + \sigma(W_{g_{cc2}}(W_{g_{cc1}}z_{dl1} + b_{g_{cc1}})) \tag{7}$$

where $W_{g_{cc1}}$ and $W_{g_{cc2}}$ are weight matrices for the GRL, $b_{g_{cc1}}$ is the bias vector, and σ is the sigmoid activation function.

The importance weights generated by the second GRL are applied to the stacked GRL outputs using matrix multiplication. This operation effectively computes a weighted sum of the transformed features, where each feature's contribution is scaled by its corresponding importance weight as follows:

$$g_{svn} = G \times g_{cc} \tag{8}$$

The output of the VSN after processing through the fully connected layer as follows:

$$h_2 = ReLU(W_2 g_{svn} + b_2) \tag{9}$$

where W_2 is the weight matrix for the output of the VSN. b_2 is the bias vector this layer.

Concatenate and Fully Connected Layer 2: After the VSN processes the inputs and generates the selected features, these features are often combined with the output of the first full connected layer before being passed through the final output layer. Specifically, the output from VSN, h_2, is concatenated with the output of the first fully connected layer, h_1, as follows:

$$z_{dl2} = Concatenate(h_1, h_2) \tag{10}$$

This newly concatenated vector is then passed through the second fully connected layer as follows:

$$\hat{y} = ReLU(W_3 z_{dl2} + b_3) \tag{11}$$

where where W_3 is the weight matrix of the second fully connected layer. b_3 is the bias vector of the second fully connected layer. And ReLU is the Rectified Linear Unit activation function.

Output: The raw output $\hat{y}$ of the preceding layer represents the unbounded predicted rating, which may lie outside the desired range. Depending on the rating scale, we need to ensure that the predicted rating falls within the valid range of [a, b]. For example, if a = 1 , b = 5 for a range of [1, 5], the model applies a clipping operation. this clipping function constrains $\hat{y}$ to be no less than 1 and no greater than 5. Therefore, the clipping operation is defined as follows:

$$\hat{y}_{clipped} = \begin{cases} a & \text{if} \quad \hat{y} < a \\ \hat{y} & \text{if} \quad a \leq \hat{y} \leq b \\ b & \text{if} \quad \hat{y} > b \end{cases} \tag{12}$$

where $\hat{y}_{clipped}$ is the final output after clipping, a, b represent the range of rating values.

With the architectural components and their interactions thoroughly explored, we conduct experiments aimed at assessing its performance across different datasets and evaluation metrics in the next section.

4 Experiments

In this section, we focus on introducing the dataset used to evaluate the performance of our approach and provide a description of the baseline models against which our approach is compared. Finally, we present our analysis of the achieved experimental results.

Table 3. Statistics on experimental datasets

Dataset	Nb of Users	Nb of Item	Nb of Contexts	Nb of Criteria	Rating Scale	Data Sparsity	Nb of Rating
ITM-Rec	454	70	3	3	[1,5]	83.54%	5230

4.1 Dataset and Evaluations

Dataset: We conduct our experiments using the ITM-Rec dataset [28]. This dataset is designed for the development and evaluation of educational recommender systems. It was collected from student questionnaires at the ITM department of Illinois Institute of Technology, USA. The dataset encompasses a variety of student evaluations based on different criteria and contexts. Table 3 summarizes the key statistics and characteristics of each dataset.

Compared Approaches: To evaluate our model's effectiveness, we compare its performance with several established baseline methods:

– **DCN** (Deep Cross Network) employs a direct approach to capturing feature interactions. By utilizing a cross network structure, it models these relationships at multiple levels of representation [27].

– **DeepFM** addresses complex feature interactions using a deep neural network. Nonlinear activations in hidden layers enable it to capture intricate patterns beyond the scope of simpler models [6].
– **xDeepFM** combines a Compressed Interaction Network (CIN) for explicit vector-wise feature interactions with a deep neural network. This hybrid model learns both bounded and unbounded feature interactions [16].
– **NFM** (Neural Factorization Machine) handles sparse data by combining the strengths of Factorization Machines (FM) and neural networks. FM captures basic feature interactions, while the neural network learns complex, higher-order relationships between features [7].
– **AutoInt** automates feature interaction learning by projecting features into multiple latent spaces. Attention mechanisms identify crucial interactions within these spaces, eliminating manual feature engineering [23].
– **FibiNET** combines feature importance and bilinear feature interactions. This model learns feature importance using a Squeeze-Excitation network (SENET) mechanism. [8].

Evaluation Metrics: We conduct our evaluation through two commonly used metrics: the Mean Absolute Error (MAE) and the Root Mean Squared Error (RMSE). Both metrics quantify the difference between the predicted values generated by the model and the actual observed values [22].

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{13}$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{14}$$

where n denotes number of observations, y_i is the actual observed value, and $\hat{y}_i$ is the predicted value by the model. In general, MAE measures the average prediction error without considering error magnitude. RMSE, on the other hand, penalizes larger errors more heavily. Both metrics assess prediction accuracy, with lower values indicating better performance. Perfect predictions result in zero for both MAE and RMSE.

Based on this diverse set of baseline approaches and two well-known evaluation metrics, we aim to provide an assessment of our model's performance and highlight its potential advantages over existing techniques. In the following sections, we will describe the results of our experiments.

4.2 Experiments Results

In this section, we present and analyze the performance of various models on the prediction task using two key metrics: RMSE and MAE. The models evaluated include **DeepFM**, **xDeepFM**, **AutoInt**, **DCN**, **NFM**, **FiBiNET**, and our proposed model (**SVN**). The results obtained are summarized in Table 4

Table 4. Performance comparison of different models based on two metrics.

Metric	DeepFM	xDeepFM	AutoInt	DCN	NFM	FiBiNET	Ours (SVN)
RMSE	0.8332	0.8265	0.8539	0.8439	0.8225	0.8366	**0.8136**
MAE	0.6364	0.6194	0.6549	0.6416	0.6173	0.6118	**0.5947**

Starting with RMSE, the **DeepFM** model achieved an RMSE of 0.8332, which is moderately competitive among the models tested. **xDeepFM** performed slightly better, with an RMSE of 0.8265, indicating an improvement in prediction accuracy. **AutoInt** and **DCN** showed relatively higher RMSE values, at 0.8539 and 0.8439, respectively, suggesting that their predictions were slightly less accurate. **NFM** demonstrated strong performance with an RMSE of 0.8225, closely rivaling the best-performing models. **FiBiNET**, with an RMSE of 0.8366, fell in the middle range. Notably, our proposed model (**SVN**) achieved the lowest RMSE of 0.8136, indicating the highest accuracy in predictions among all the models tested in this experiment.

Turning to MAE, **DeepFM** resulted in an MAE of 0.6364, which is relatively higher compared to the other models, indicating larger average errors. **xDeepFM** showed improvement with an MAE of 0.6194, while **AutoInt** had the highest MAE at 0.6549, suggesting that it made the most significant errors on average. **DCN** and **NFM** exhibited competitive MAE values at 0.6416 and 0.6173, respectively, indicating consistent performance across these models. **FiBiNET** slightly outperformed others with an MAE of 0.6118. Our proposed model, once again, demonstrated superior performance by achieving the lowest MAE at 0.5947, highlighting its ability to minimize errors effectively.

Comparing the models, it is evident that our proposed **SVN** model consistently outperformed the others across both metrics. With the lowest RMSE (0.8136) and MAE (0.5947), **SVN** demonstrated the best predictive accuracy and the smallest average error. This suggests that the **SVN** architecture effectively captures the necessary features and interactions, leading to improved prediction performance. While **xDeepFM** and **NFM** were the closest competitors, they did not match the performance of the **SVN** model, indicating that the additional complexity and feature selection capabilities of **SVN** offer a tangible advantage. In contrast, **AutoInt** and **DCN** underperformed relative to the other models, especially in terms of MAE, indicating that their model structures might be less effective in minimizing average errors. **FiBiNET** and **DeepFM** also provided competitive results but fell short of the performance of the **SVN** model, suggesting that while they capture interactions effectively, they may lack the adaptive feature selection or other innovations present in **SVN**.

Overall, the experimental results demonstrate that our approach based on SVN model outperforms established models in both RMSE and MAE on the ITM-Rec dataset. This superior performance can be attributed to the models ability to dynamically select and weight the most relevant features, leading to more accurate and reliable predictions. The success of the SVN model under-

scores the importance of adaptive feature selection in enhancing model performance for complex tasks.

5 Conclusion

In this paper, we investigated the development of a context-aware multi-criteria recommender system using a variable selection network. Leveraging VSN, our approach employs dynamic feature selection through a series of fully connected layers that refine the features and generate the final predictions. This structure allows the model to adapt to various contexts and criteria, making it highly effective for complex recommendation tasks. Our approach was evaluated against several leading baselines using RMSE and MAE as evaluation metrics. The experimental results indicate that our approach based on SVN outperforms these established methods, achieving the highest accuracy and lowest error rates across all tested scenarios. This improvement can be attributed to the SVN's ability to dynamically prioritize and integrate the most pertinent features. The findings from this research underscore the potential of adaptive feature selection in advancing the field of recommender systems, offering new avenues for more precise and contextually relevant recommendations. Moving forward, we aim to enhance the integration of complex contextual factors, improve scalability and efficiency, and increase the explainability of deep learning models in context-aware MCRSs. Additionally, exploring hybrid models with technologies like reinforcement learning and graph neural networks could open up new research opportunities.

References

1. Adomavicius, G., Kwon, Y.: New recommendation techniques for multicriteria rating systems. IEEE Intelligent Systems **22**(3), 48–55 (2007)
2. Afzal, I., Yilmazel, B., Kaleli, C.: An approach for multi-context-aware multi-criteria recommender systems based on deep learning. IEEE Access (2024)
3. Alamdari, P.M., Navimipour, N.J., Hosseinzadeh, M., Safaei, A.A., Darwesh, A.: A systematic study on the recommender systems in the e-commerce. Ieee Access **8**, 115694–115716 (2020)
4. Batmaz, Z., Kaleli, C.: Ae-mccf: an autoencoder-based multi-criteria recommendation algorithm. Arabian Journal for Science and Engineering **44**, 9235–9247 (2019)
5. Dridi, R., Tamine, L., Slimani, Y.: Exploiting context-awareness and multi-criteria decision making to improve items recommendation using a tripartite graph-based model. Information Processing & Management **59**(2), 102861 (2022)
6. Guo, H., Tang, R., Ye, Y., Li, Z., He, X.: Deepfm: a factorization-machine based neural network for ctr prediction. arXiv preprint arXiv:1703.04247 (2017)
7. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.S.: Neural collaborative filtering. In: Proceedings of the 26th international conference on world wide web. pp. 173–182 (2017)
8. Huang, T., Zhang, Z., Zhang, J.: Fibinet: combining feature importance and bilinear feature interaction for click-through rate prediction. In: Proceedings of the 13th ACM conference on recommender systems. pp. 169–177 (2019)

9. Jannach, D., Zanker, M., Fuchs, M.: Leveraging multi-criteria customer feedback for satisfaction analysis and improved recommendations. Information Technology & Tourism **14**, 119–149 (2014)
10. Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 426–434 (2008)
11. Le, N.L., Abel, M.H., Gouspillou, P.: Towards an ontology-based recommender system for the vehicle domain. In: 3rd International Conference on Deep Learning, Artificial Intelligence and Robotics,(ICDLAIR). vol. 441, pp. 107–116 (2021)
12. Le, N.L., Abel, M.H., Gouspillou, P.: Combining embedding-based and semantic-based models for post-hoc explanations in recommender systems. In: 2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC). IEEE (2023)
13. Le, N.L., Abel, M.H., Gouspillou, P.: A constraint-based recommender system via rdf knowledge graphs. In: 2023 26th International Conference on Computer Supported Cooperative Work in Design (CSCWD). pp. 849–854. IEEE (2023)
14. Le, N.L., Abel, M.H., Gouspillou, P.: A personalized recommender system based-on knowledge graph embeddings. In: The 3rd International Conference on Artificial Intelligence and Computer Vision (AICV2023), March 5–7, 2023. pp. 368–378. Springer Nature Switzerland, Cham (2023)
15. Le, N.L., Zhong, J., Negre, E., Abel, M.H.: Constraint-based recommender system for crisis management simulations. arXiv preprint arXiv:2306.04553 (2023)
16. Lian, J., Zhou, X., Zhang, F., Chen, Z., Xie, X., Sun, G.: xdeepfm: Combining explicit and implicit feature interactions for recommender systems. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining. pp. 1754–1763 (2018)
17. Lim, B., Arık, S.Ö., Loeff, N., Pfister, T.: Temporal fusion transformers for interpretable multi-horizon time series forecasting. International Journal of Forecasting **37**(4), 1748–1764 (2021)
18. Manouselis, N., Costopoulou, C.: Analysis and classification of multi-criteria recommender systems. World Wide Web **10**, 415–441 (2007)
19. Nassar, N., Jafar, A., Rahhal, Y.: A novel deep multi-criteria collaborative filtering model for recommendation system. Knowledge-Based Systems **187**, 104811 (2020)
20. Obeid, C., Lahoud, I., El Khoury, H., Champin, P.A.: Ontology-based recommender system in higher education. In: Companion proceedings of the the web conference 2018. pp. 1031–1034 (2018)
21. Shambour, Q.: A deep learning based algorithm for multi-criteria recommender systems. Knowledge-based systems **211**, 106545 (2021)
22. Shani, G., Gunawardana, A.: Evaluating recommendation systems. Recommender systems handbook pp. 257–297 (2011)
23. Song, W., Shi, C., Xiao, Z., Duan, Z., Xu, Y., Zhang, M., Tang, J.: Autoint: Automatic feature interaction learning via self-attentive neural networks. In: Proceedings of the 28th ACM international conference on information and knowledge management. pp. 1161–1170 (2019)
24. Tallapally, D., Sreepada, R.S., Patra, B.K., Babu, K.S.: User preference learning in multi-criteria recommendations using stacked auto encoders. In: Proceedings of the 12th ACM conference on recommender systems. pp. 475–479 (2018)
25. Vu, S.L., Le, Q.H.: A deep learning based approach for context-aware multi-criteria recommender systems. Comput. Syst. Sci. Eng. **44**(1), 471–483 (2023)
26. Wang, J.: Multi-criteria recommender system with hybrid deep tensor decomposition. In: Proceedings of the 2021 4th International Conference on Data Storage and Data Engineering. pp. 65–69 (2021)

27. Wang, R., Fu, B., Fu, G., Wang, M.: Deep & cross network for ad click predictions. In: Proceedings of the ADKDD'17, pp. 1–7 (2017)
28. Zheng, Y.: Itm-rec: An open data set for educational recommender systems. arXiv preprint arXiv:2303.10230 (2023)
29. Zheng, Y., Shekhar, S., Jose, A.A., Rai, S.K.: Integrating context-awareness and multi-criteria decision making in educational learning. In: Proceedings of the 34th ACM/SIGAPP symposium on applied computing. pp. 2453–2460 (2019)