DEEP LEARNING-BASED MULTI-CONTEXT-AWARE MULTI-CRITERIA RECOMMENDER SYSTEM

DOCTORAL THESIS

Ifra AFZAL

Eskişehir 2024

DEEP LEARNING-BASED MULTI-CONTEXT-AWARE MULTI-CRITERIA RECOMMENDER SYSTEM

Ifra AFZAL

PhD Dissertation

Department of Computer Engineering

Programme in Computer Science

Supervisor: Prof. Dr. CİHAN KALELİ

(Co-Supervisor: Asst. Prof. Dr. BURCU YILMAZEL)

Eskişehir

Eskişehir Technical University

Institute of Graduate Programs

August 2024

FINAL APPROVAL FOR THESIS

This thesis titled DEEP LEARNING-BASED MULTI-CONTEXT-AWARE MULTI-CRITERIA RECOMMENDER SYSTEM has been prepared and submitted by Ifra AFZAL in partial fulfillment of the requirements in "Eskişehir Technical University Directive on Graduate Education and Examination" for the Degree of PhD in Computer Engineering Department has been examined and approved on 09/08/2024.

Committee Members		Title, Name and Surname	Signature
Member	:	Prof. Dr. Cihan KALELİ (Supervisor)	
Member	:	Prof. Dr. Serkan GÜNAL	
Member	:	Assoc. Prof. Dr. Alper BİLGE	
Member	:	Assist. Prof. Dr. Zühal KARTAL	
Member	:	Assist. Prof. Dr. Tuğba TÜRKOĞLU KAYA	

Prof. Dr. Semra KURAMA
Director of the Institute of Graduate Programs

09/08/2024

SUPERVISOR APPROVAL

PhD student Ifra AFZAL, whom I supervise, has completed this thesis titled DEEP LEARNING-BASED MULTI-CONTEXT-AWARE MULTI-CRITERIA RECOMMENDER SYSTEM. According to my inspections, the work is scientifically and ethically appropriate for the student to the thesis defense exam.

Supervisor
Prof. Dr. CİHAN KALELİ

ABSTRACT

DEEP LEARNING-BASED MULTI-CONTEXT-AWARE MULTI-CRITERIA RECOMMENDER SYSTEM

Ifra AFZAL

Department of Computer Engineering

Programme in Computer Science

Eskişehir Technical University, Institute of Graduate Programs, August 2024

Supervisor: Prof. Dr. CİHAN KALELİ

(Co-Supervisor: Asst. Prof. Dr. BURCU YILMAZEL)

As digital information becomes ubiquitous, Recommender Systems (RSs) become increasingly critical in navigating this vast landscape. In this dissertation, we propose a novel deep learning-based approach to integrate multi-context and multi-criteria data into a unified neural network model. A context-aware and multi-criteria recommender system enhances traditional two-dimensional methods of RSs by incorporating context awareness and multiple criteria. In our work, data from multiple contexts and multiple criteria are intricately woven together within a single architecture, unlike traditional approaches. Concurrent processing allows sophisticated interactions between context and criteria, resulting in more accurate recommendations. When context-aware systems make recommendations, they incorporate contextual factors, such as time and location, while multi-criteria-based approaches utilize a broader range of evaluative criteria and offering more relevant suggestions. For making recommendations, contextual information and multi-criteria ratings have not been used together, despite both approaches having advantages in producing more accurate and personalized referrals. To fill this gap, a context-aware, multi-criteria recommender system is needed. We employ a deep learning model that integrates context-aware recommendation systems with multi-criteria recommendation systems, instead of processing each separately. Hence the system becomes more adaptive and delivers more accurate recommendations, as contextual and criteria-specific insights are combined to create personalized recommendations. The proposed method provides more accurate predictions than state-of-the-art recommendation techniques when applied to TripAdvisor and ITM-Rec datasets.

Keywords: Context-aware, Deep learning, Multi-criteria, Recommender systems.

ÖZET

DERİN ÖĞRENME TABANLI ÇOKLU BAĞLAMA DUYARLI ÇOK KRİTERLİ ÖNERİ SİSTEMİ

Ifra AFZAL

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Bilimleri Bilim Dalı

Eskişehir Teknik Üniversitesi, Lisansüstü Eğitim Enstitüsü, Ağustos 2024

Danışman: Prof. Dr. CİHAN KALELİ

İkinci Danışman: Dr. Öğr. Üyesi BURCU YILMAZEL

Dijital bilginin yaygın hale gelmesiyle birlikte, Öneri Sistemleri (ÖS), bu geniş bilgi evreninde yol alabilmek için giderek daha kritik bir rol üstlenmektedir. Bu doktora tezinde, çoklu bağlam ve çoklu kriter verilerini tek bir sinir ağı modeli altında entegre eden yenilikçi bir derin öğrenme yaklaşımı önerilmektedir. Bağlam farkındalığına sahip ve çoklu kriterli bir öneri sistemi, geleneksel iki boyutlu ÖS yöntemlerine göre, bağlam farkındalığını ve birden fazla kriteri bünyesinde barındırarak önemli bir gelişme sunmaktadır. Çalışmamızda, çoklu bağlam ve kriterlerden elde edilen veriler, geleneksel yaklaşımlardan farklı olarak, tek bir yapı içerisinde işlenmektedir. Eşzamanlı işleme, bağlam ve kriterler arasında gelişmiş etkileşimlere olanak tanımakta ve böylece daha isabetli öneriler elde edilmektedir. Bağlam farkındalığına sahip sistemler, önerilerde bulunurken zaman ve konum gibi bağlamsal faktörleri dikkate alırken, çoklu kriter tabanlı yaklaşımlar, daha geniş bir değerlendirme ölçütü yelpazesi sunarak daha ilgili öneriler geliştirmektedir. Her iki yaklaşım da daha doğru ve kişiselleştirilmiş öneriler sağlama potansiyeline sahip olmasına rağmen, bağlamsal bilgi ve çoklu kriter değerlendirmeleri birlikte kullanılmamıştır. Bu eksikliği gidermek adına, bağlam farkındalığına sahip, çoklu kriterli bir öneri sistemine ihtiyaç duyulmaktadır. Bu doğrultuda, her bir yaklaşımı ayrı ayrı ele almak yerine, bağlam farkındalığına sahip öneri sistemlerini çoklu kriter öneri sistemleri ile entegre eden bir derin öğrenme modeli geliştirilmiştir. Böylelikle sistem, bağlamsal ve kriterlere özgü değerlendirmeleri birleştirilmesiyle daha uyarlanabilir hale gelmekte ve daha doğru öneriler sunmaktadır. Önerilen yöntem, TripAdvisor ve ITM-Rec veri kümeleri üzerinde uygulandığında, güncel öneri tekniklerinden daha yüksek doğruluk oranı sağlamaktadır.

Anahtar Sözcükler: Bağlama-duyarlı, Derin öğrenme, Çok kriterli, Öneri sistemleri.

ACKNOWLEDGEMENTS

Firstly, I am deeply grateful to Almighty Allah, the Most Beneficent and Merciful, for His blessings.

I am genuinely grateful to my supervisor, Prof. Dr. **Cihan KALELİ** and, co-supervisor, Assist. Prof. Dr. **Burcu YILMAZEL**, for their support and direction during my research. It would not have been possible without their assistance. I consider it a great honor for me to work under their supervision.

I would like to acknowledge the appreciation and constructive feedback provided by Prof. Dr. **Serkan GÜNAL** and Assist. Prof. **Dr. Zühal KARTAL** during the committee meetings, which motivated me to work harder.

Additionally, I am very grateful to Assoc. Prof. Dr. **Alper BİLGE** and Assist. Prof. Dr. **Tuğba TÜRKOĞLU KAYA** for their valuable contributions to my dissertation committee.

I am deeply grateful for the love, support and prayers of my parents, especially my beloved father **Muhammad AFZAL**, as well as my siblings, throughout my life. I also appreciate the unwavering support and tolerance of my partner, **Mansab Ali MOON**, during my research studies. And, a special note of love and thanks goes to my beloved son **ABDURRAHMAN** who cheers me up every morning with his sweet and bright smile.

Lastly, I am greatly indebted to to the **TÜRKİYE BURSLARI** program, funded by the Government of Turkey, for providing me the financial support needed to complete my doctoral studies. This invaluable platform made my PhD journey possible.

Ifra AFZAL

09/08/2024

STATEMENT OF COMPLIANCE WITH ETHICAL PRINCIPLES AND RULES

I hereby truthfully declare that this thesis is an original work prepared by me; that I have behaved in accordance with the scientific ethical principles and rules throughout the stages of preparation, data collection, analysis and presentation of my work; that I have cited the sources of all the data and information that could be obtained within the scope of this study, and included these sources in the references section; and that this study has been scanned for plagiarism with "scientific plagiarism detection program" used by Eskişehir Technical University, and that "it does not have any plagiarism" whatsoever. I also declare that, if a case contrary to my declaration is detected in my work at any time, I hereby express my consent to all the ethical and legal consequences that are involved.

Ifra AFZAL

CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

Page

GLOSSARY OF SYMBOLS AND ABBREVIATIONS

AE	:	Auto Encoder
AEMC	:	Auto Encoder Multi-Criteria
AHP	:	Analytical Hierarchy Process
ANFIS	:	Adaptive Neuro-Fuzzy Inference Systems
CAMF	:	Context-Aware Matrix Factorization
CAMF-C	:	Context-Aware Matrix Factorization Context
CAMF-CI	:	Context-Aware Matrix Factorization Context-Item
CAMF-CU	:	Context-Aware Matrix Factorization Context-User
CARS	:	Context-Aware Recommender System
CB	:	Content Based
CF	:	Collaborative Filtering
CMRS	:	Context-Aware Multi-Criteria Recommender System
CNN	:	Convolutional Neural Network
CRESC	:	Contextual Rating Enriched by Similar Communities
CRMC	:	Contextual Rating Multi-Criteria Communities
DA	:	Data Analytics
DB	:	Database
DL	:	Deep Learning
DL-CARS	:	Deep Learning based Context-Aware Recommender System
DL-MCRS	:	Deep Learning based Multi-Criteria Recommender System
DNNs	:	Deep Neural Networks
DS	:	Data Science
FM	:	Factorization Machine
FNB	:	Fuzzy Naïve Bayesian
GNN	:	Graph Neural Network
GP	:	Genetic Programming
HOSVD	:	Higher-Order Singular Value Decomposition
IBCF	:	Item-Based Collaborative Filtering
IMCCF	:	Item-Based Multi-Criteria Collaborative Filtering
IR	:	Information Retrieval
LR	:	Linear Regression

MAE	:	Mean Absolute Error
MC-CF	:	Multi-Criteria Collaborative Filtering
MCoMC	:	Multi-Context Multi-Criteria
MCoMCRS	:	Multi-Context Multi-Criteria Recommender System
MCRS	:	Multi-Criteria Recommender System
MD	:	Multi-Dimensional
MF	:	Matrix Factorization
ML	:	Machine Learning
MLP	:	Multi Layer Perceptron
NCF	:	Neural Collaborative Filtering
POI	:	Point of Interest
ReLU	:	Rectified Linear Unit
RMSE	:	Root Mean Square Error
RNN	:	Recurrent Neural Network
RSs	:	Recommender Systems
SCoMCRS	:	Single-Context Multi-Criteria Recommender System
SCoRS	:	Single-Context Recommender System
SVD	:	Singular Value Decomposition
TF	:	Tensor Factorization
UBCF	:	User-Based Collaborative Filtering
UTAUT	:	Unified Theory of Acceptance and Use of Technology

1. INTRODUCTION

In the decade of big data, the World Wide Web has been expanding dramatically many websites (including Google, Yahoo!, Amazon, YouTube, Netflix, and others) have steadily grown in size and complexity, making it harder and longer for their users to find the items that they want. User-specific suggestions for items they might like are provided by RSs (Gunes et al., 2014). At the beginning of the 90s, RSs were initially contrived (Goldberg et al., 1992), and are now considered to be part of an independent research field from Information Retrieval (IR) (Herlocker et al., 2000). The importance of recommender systems cannot be denied, as they are considered an integral tool in filtering online content. These systems have evolved from a luxury to a necessity. RSs analyze users' preferences and behaviors to help them find relevant goods, services, and content. It has been widely used in a variety of applications. For example, movies (Netflix), e-news (Yahoo! News Today), music (Spotify), e-commerce (Amazon.com), social networks (Facebook), tourism (TripAdvisor), etc. Traditional recommendation systems primarily employ a variety of approaches to predict user preferences, including Content-Based (CB) filtering, Collaborative Filtering (CF), knowledge-based methods, demographic-based methods, and hybrid recommendation approaches (Manouselis et al., 2010),(Bobadilla et al., 2013). In the following section, a brief introduction and challenges of these RSs are discussed.

1.1. Traditional Types of Recommender System

Recommender systems can be categorized into various types based on the methodological approaches utilized to generate recommendations. A visual overview of the RS types is presented in **Figure *1.1***.

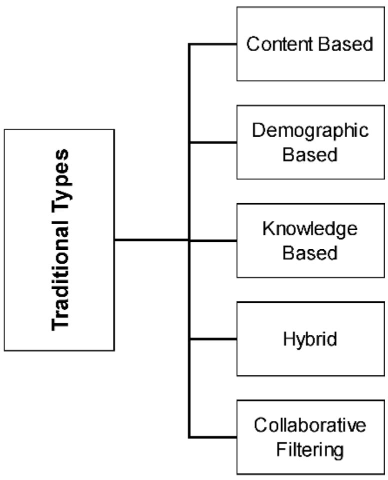

Figure 1.1. *Types of recommender systems*

1.1.1. Content-based recommendation systems

Content-based recommendation systems suggest products based on the features of the items and the user's past activities with similar items. These systems analyze the content of the items and compare it to a user's profile, which is composed of their preferences and past actions (Lops et al., 2011). For instance, a news recommendation system that suggests articles to a user based on their previous reading and preference history. This is done by analyzing the relevant keywords and topical content.

1.1.2. Collaborative filtering

Collaborative filtering systems make recommendations by investigating patterns of user behavior such as ratings, clicks or purchases histories in order to identify similarities between users or items (Schafer et al., 2007). There are two primary types of collaborative filtering: 1) User-based Collaborative Filtering (UBCF) and 2) Item-based Collaborative Filtering (IBCF). UBCF methods recommend items that have been preferred by users with similar preferences while IBCF methods recommend items similar to those a user has liked previously.

1.1.3. Knowledge -based recommendation systems

Knowledge-based RSs leverage domain expertise about users and items to provide personalized recommendations (Burke, 2013). These systems often employ rule-based approaches or case-based reasoning to align user needs with product features. For instance, a financial services recommendation system may suggest investment products tailored to a user's financial objectives and risk profile.

1.1.4. Demographic -based recommendation systems

Demographic-based RSs leverage user demographic data (age, sex, income, education etc.,) to categorize users and provide recommendations tailored to these demographic groupings (Safoury & Salah, 2013). This approach assumes that users with similar demographic data have similar preferences. For instance, an advertising companies that targets ads according to user demographics information.

1.1.5. Hybrid recommendation systems

Hybrid recommendation systems integrate multiple recommendation techniques to capitalize on their respective advantages and offset their individual limitations. These hybrid techniques can combine content-based RS with CF, or knowledge-based RSs.

1.2. Challenges and Issues in Traditional Recommendation Systems

Following are some commonly found challenges in traditional recommender system.

1.2.1. Cold start problem

Cold start problems occur when new users or items are added to the system. As a result, it becomes more difficult to predict either the taste or the purchase behavior of new users, which results in less accurate recommendations. A number of solutions can be found to resolve the cold-start problem, including asking the user to rate certain items at the beginning, asking the users to express their preferences, or suggesting items based on the collected demographic information. Traditional recommendation system approaches are unable to accomplish this due to the inherent limitations of their underlying methodologies (Lika et al., 2014). The employment of advanced recommendation system techniques, like context-aware recommender Systems and multi-criteria recommender system, which leverage more comprehensive user data, has resulted in enhanced recommendation accuracy (Vu & Le, 2023).

1.2.2. Privacy

Providing personal information to recommender systems leads to better recommendation services, but may pose privacy and security risks. The problem of data privacy makes users uneasy about feeding data into recommender systems. Therefore, a recommender system should create trust among its users. Traditional recommender systems such as Collaborative Filtering are more prone to such issues because user data is stored in a central repository which may cause misuse of data (Shao & Xie, 2019), (Polat & Du, 2005).

1.2.3. Sparsity

A large amount of data about items in the catalog and the lack of willingness of users to rate items lead to a dispersed profile matrix, creating less accurate recommendations. Items are difficult to predict accurately because of the sparse rating. It can be extremely challenging to provide accurate recommendations for very few rated items that have multidimensional vectors (Y. Zhang & Wang, 2010). There are several approaches to deal with this situation, including multidimensional recommendation models (Batmaz & Kaleli, 2019). But this problem still needs to be addressed because, with the growing use of social media, data is becoming more sparse.

1.2.4. Scalability

A scalability issue arises when the number of users and items in the system exceeds the performance limits of traditional recommendation algorithms. As a result, computational resources needed for processing become exceedingly high. To maximize scalability, the original user-item rating matrix can be expanded to include additional information such as user profiles, user social network activities, etc., which would then enable the use of more advanced algorithms (Unger et al., 2020).

1.2.5. Shilling attacks

Shilling attacks occur when a malicious user or a competitor enters in a system and starts giving false ratings to the items. The purpose of giving false ratings to some items is to either increase or decrease their popularity (F. Zhang, 2010). RSs are vulnerable to these attacks which can damage their trustworthiness as well as reduce the recommendation quality and performance. CF techniques are a significant area of concern in shilling attacks. Several methods can be employed to identify malicious activities and a range of techniques can be used to identify and mitigate attacks (Gunes et al., 2014).

1.3. Problem Statement

The traditional recommendation system, which is a two-dimensional recommender system, generates a list of suggestions based on user preferences. However, this approach has limitations as it only considers 2D of information about users and items. Moreover, traditional recommendation algorithms usually have cold start, data sparsity and scalability problems resulting in incorrect predictions on top of the recommendations suggested. Therefore, the accuracy of user recommendations is still a major academic problem to improve. During the past decade, there have been recommended techniques proposed by researchers to solve this issue. Multi-Criteria Recommender Systems (MCRS) and Context-Aware Recommender System (CARS) suggest few of such approaches that improved recommendation performance. MCRSs improve suggestions by taking into account multiple user preferences while CARSs incorporate contextual information such as time, location, and mood. Both individual RS approaches have shown promising results in many studies (Santana & Domingues, 2020), (Bilge & Kaleli, 2014). However, integrating these approaches is difficult and presents an opportunity for innovation despite the best of their capabilities. In the problem statement we investigate how MCRS and CARS can be integrated in single framework: Multi-context Multi-

Criteria Recommender System (MCoMCRS). The study uses Deep Neural Networks (DNNs) to improve the precision of recommendations.

The efficiency of the model is a critical concern. The traditional practice of using separate models to integrate CARS and MCRS suffers in efficiency (Vu & Le, 2023). Most of the above-mentioned models are Machine Learning (ML) based. So, the implementation of separate models for CARS and MCRS is a complex process in these models (Zheng et al., 2019). Traditionally, ML algorithms deal with feature extraction and classification separately. Therefore, to train, maintain and test two separate models often increase the computational cost of these approaches. Also, using separate models affects the ability to understand the primary motivation behind the system's recommendations. Considering all these limitations, we have proposed a unified model to handle both types of data (multi-context data and multiple criteria ratings), which reduces the computational burden and simplifies deployment.

An important limitation of existing integrated MCRS and CARS studies is their inability to fully capture the complete context of user needs (Rodríguez & Viktor, 2013), (Vu & Le, 2023), (Ramchandani et al., 2020). A single context, such as location or time, is typically examined in these studies while other contexts which can play a significant role in recommendations are ignored. This narrow focus can significantly impair the quality of recommendations. For instance, a user looking for a movie to watch on a weekend. A location-based CARS may only suggest nearby theaters, rather than suggesting a family friend's movie based on the user's current contextual preference.

Existing integration also faces the challenge of merging the diverse data requirements of CARS and MCRS. However, handling large-scale data efficiently is crucial, particularly when there is a need to process multi-context and multiple criteria information simultaneously. Moreover, the contextual embedding process is also important in CARS, as analyzing and evaluating contextual information is challenging. Similarly, making recommendations is difficult when the data is multi-dimensional and complex due to the presence of multiple ratings in MCRS.

The contribution of this thesis is to develop a unified DNN that integrates the processes of context embedding and recommendation generation that based on multiple ratings with accurate results. Through a DNN architecture, the MCoMCRS framework addresses above mentioned limitations. There are several advantages to this unified model that are discussed in Dissertation Contribution Section. We will learn more about the problem in the coming chapters.

1.4. Research Questions

The vast volume of data available on the internet has undoubtedly become a challenge for conducting effective and relevant searches. Online activities have become a fundamental necessity, as exemplified by the increased preference for online shopping during the COVID-19 pandemic. Consequently, the need for more accurate and relevant item recommendations on the internet has become a ubiquitous aspect of daily life. To address this issue, we have proposed a novel method based on Deep Learning (DL), MCoMCRS, which integrates well-known approaches in the field of recommender systems, CARS and MCRS. The key research questions are:

i. Does the proposed model deliver improved performance outcomes compared to the non-contextual RS?

ii. Does our approach more effective than MCRS based on ML technique?

iii. Is our proposed approach superior to single-context and multi-context CARS techniques that typically use traditional ML algorithms?

iv. Does our unified DNN model outperformed when it compared to individual DL-based CARS and DL-based MCRS models?

v. Additionally, is the performance of MCoMCRS more favorable compared to the DL-based single-context with MCRS?

vi. How can the DNN model be trained efficiently given the large and potentially sparse datasets typical of recommender systems?

vii. Is this unified DNN model overfitted to large complex datasets?

viii. How can the robustness of the unified DNN model be ensured when handling multi-context and multi-criteria data?

To answer these questions, we need to compare the performance of the proposed MCoMCRS system with traditional and existing recommendation systems.

1.5. Dissertation Contributions

A novel DL-based approach proposed by (Afzal et al., 2024) integrates multiple contextual conditions and multi-criteria ratings. This advanced unification allows us to leverage multiple contextual features and preference criteria to provide more personalized recommendations.

The study utilized a single DNN model to more accurately predict the overall rating. We evaluate the performance of our approach with a real-world dataset in comparison to state-of-the-art methods published in related literature. This rigorous evaluation shows the benefits and performances of the proposed method. Also, we perform a comparative analysis of our MCoMCRS approach with a single context-based MCRS. The results indicate the importance of using multiple contextual conditions in recommendation tasks. The contribution of the dissertation is discussed as follows.

First, the proposed MCoMCRS obtains a deeper understanding of user needs by considering information from multiple contexts, allowing it to provide more personalized recommendations. Second, comparing the MCoMCRS to discrete models of CARS and MCRS, achieves efficient learning and improves interpretability. This integrated framework simplifies the broader system design and enhances the understanding of how multiple contextual conditions and different user preferences influence the final recommendations. Additionally, this proposed method mitigates the computational cost compared to developing and maintaining two unconnected model for CARS and MCRS.

The concept of integration is obtained from traditional ML algorithms, where feature extraction and integration are separate tasks. DNNs perform these tasks as one operation using complex data from various sources. It can capture intricate relationships between multi-criteria ratings and contexts through extensive training. In other words, a single DNN model handles both multi-contextual and multi-criteria information to deliver personalized content seamlessly.

Our model has exhibited adaptability across an array of applications involving multidimensional user preferences. This is evidenced by the successful implementation and assessment of the model on multiple data sources such as travel and educational domains, underscoring the model's robustness and adaptability.

Various studies (Yeung et al., 2012), (Gasparic & Ricci, 2015), (Nguyen & Nguyen, 2015), (Ashley-Dejo et al., 2016), (Toledo et al., 2019), (Ramchandani et al., 2020), (Sassi et al., 2021), (Zheng, 2019a), (Zheng et al., 2019), (Dridi et al., 2022), (Dridi et al., 2019), (Rodríguez & Viktor, 2013), (Vu & Le, 2023) explored combining CARS and MCRS. However, most research focused on single contexts and primarily used traditional ML methods. This indicates to further explore the integration of CARS and MCRS using DL. To date, no literature review has analyzed the integration of Context-aware Multi-criteria Recommender Systems (CMRS). This study also aims to address this gap by providing a targeted examination of CMRS as discussed in Chapter 3. It emphasizes how integrating context awareness and multi-criteria approaches can meaningfully contribute in the improvement of recommendation accuracy.

1.6. Organization of the Dissertation

Chapter 2 explains the background of context-aware recommender systems and multi-criteria recommender systems. Chapter 3 presents a detailed description and analysis of the integration of the context-aware multi-criteria recommender system. The deep learning-based multi-context aware multi-criteria recommender system proposed in this dissertation is thoroughly detailed in Chapter 4. Chapter 5 presents the experimental evaluation of the proposed technique, and Chapter 6 discusses the results, analysis, and insights gained from this technique. Finally, the dissertation concludes with an exploration of future research directions in Chapter 7.

2. FUNDAMENTALS OF CONTEXT-AWARE RECOMMENDER SYSTEM AND MULTI-CRITERIA RECOMMENDER SYSTEM

The two primary approaches in the field of recommender systems are context-aware recommender systems and multi-criteria recommender systems. It is essential to begin with a discussion of these approaches in order to understand how they can be integrated. This chapter presents a detailed examination of CARS and MCRS.

2.1. Context-Aware Recommender System

(Abowd et al., 1999) defined “context” as any information that can be used to characterize an object’s situation. So, CARS takes advantage of contextual situations (such as time, weather, and location) that influence user preferences to optimize the value of recommendations. RSs need to be aware of context because user preferences vary according to the context. Contextual information was originally added as additional information to recommender systems (Adomavicius & Tuzhilin, 2005a), and they demonstrated how it significantly impacted the effectiveness of recommendations. Up to now, CARS has played a significant role in recommendation systems and considerably improved RS performance in various domains. After adding context into two-dimensional RS, the rating function of CARS becomes,

$$R\text{: } User \times Item \times C_1 \times C_2 \times C_3 \times \ldots . \times C_n \rightarrow Rating \tag{2.1}$$

Where C_1 to C_n indicate contextual conditions.

In order to clarify how contextual factors, affect recommendations, let's take a look at an example. There are two users U1 and U2, two restaurants R1 and R2, and three different contexts with their contextual conditions. The location (home, work), weather (sunny, cloudy, rainy, snowing), and mood (happy, sad, angry, surprise) are presented in **Table *2.1***. In different contexts, the same user provides different ratings for the same restaurants. It is clear from **Table *2.1*** that context is important for suggestions. For example, U1 rates R1 differently based on context (Home, Sunny, Happy) than context (Work, Snowing, Anger), even if it is the same restaurant. It is also the case with U2. We can see from these illustrations that contextual information can significantly affect the results of recommendations because users need to be provided relevant recommendations according to their location, time, current activity, emotional state and past behaviors.

Table 2.1. *Contextual rating on restaurant*

User	Item	Rating	Mood	Location	Weather
U1	R1	2	Happy	Home	Sunny
U2	R2	5	Sad	Work	Cloudy
U1	R1	4	Angry	Work	Snowing
U2	R2	3	Surprise	Home	Rainy

The integration of contextual information is vital for enhancing the functionality and efficacy of recommender systems. By incorporating contextual data, CARS can deliver highly personalized, relevant, and timely recommendations. Following is a discussion of how context is important in the RS.

Incorporating contextual information enables CARS to deliver highly personalized recommendations that are more relevant and appealing to users based on their current situation. For instance, a user may prefer different types of books when traveling in the morning as compared to reading at home at night.

Contextual information in RS supports users to make more informed decisions by offering suitable options for their circumstances. For example, based on the user's fitness level, he/she could be advised to eat lighter meals during lunchtime or more intensive workout routines.

Context factors can also help to determine and filter out those recommendations that are unlikely, inappropriate, or irrelevant for a specific context. For example, to suggest outdoor activities, but not in the case of bad weather.

To better understand the usually complex and multi-faceted demands of users, contextual information is essential. In fact, by considering a variety of context related factors, CARS creates recommendations that are more suited to user-specific needs.

Moreover, the use of contextual information gives a more detailed perspective towards how users are behaving which might not look obvious in traditional recommendation techniques.

The context allows recommender systems to adapt dynamically to changes in user behavior and the environment. Adapting recommendations in real-time ensures that they remain relevant as the user's situation changes. For example, an event or restaurant that is near your location can be suggested through a location-based service.

For new users with limited interaction history, contextual data can be pivotal in generating initial recommendations. For instance, location-based suggestions can be provided even when the user has not disclosed substantial personal information. Contextual factors assist in addressing the cold start challenge by leveraging external contextual data, such as current geographic location or device type.

After understanding the context and its importance, it is necessary to know the types of contexts because in the recommendation process this categorization help us how to use contextual information under different circumstances. According to changing behavior, context can be categorized as static and dynamic. While one the basis of how much a recommender system know about context it can be classified as fully observable, partially observable and unobservable context (Zheng & Mobasher, 2018).

In the development of a recommender system, the first step is to collect contextual data. The process of gathering contextual information is referred as context acquisition. This can be done using explicit, implicit, and inferential methods (Suhaim & Berri, 2021). The explicit context acquisition is directly by the user e.g., asking questions to the users. On the other hand, the implicit acquisition consists of identifying and extracting hidden information or factors that influence the user. For instance, mobile devices constantly obtain user location. The majority of implicit factors are dynamic. Therefore, it is difficult to handle them and also have significant impact on the performance of recommendation system. To acquire the context information through inference, advanced approaches including data mining, statistical computations, and artificial intelligence algorithms are used (Salau et al., 2022).

When the context acquisition process is done the next step is how and when to use this context in recommendation process. The context-aware recommendation contains three different paradigms for integration, including pre-filtering, post-filtering, and contextual modeling (Campos et al., 2013), (Haruna et al., 2017). In pre-filtering (shown in **Figure *2.1***), the dataset is filtered before applying conventional recommendation algorithms with the help of contextual information. (Adomavicius & Tuzhilin, 2005) is an illustration of prefiltering in which all prioritized suggestions are calculated using traditional techniques such as CF, and then, based on contextual information, they are re-ranked for each user. In post-filtering (shown in **Figure *2.2***), a conventional 2D recommendation system is used to predict the ratings by ignoring the contextual information initially, adjusting as the last step the resulting set of recommendations for each user by using contextual information (Panniello et al., 2009). The contextual modeling (shown in **Figure *2.3***), directly incorporates contextual information as a significant component of the 2D recommendation system. Pre-filtering and contextual modeling are the most popular and effective methods of building CARS, while post-filtering is rarely researched.

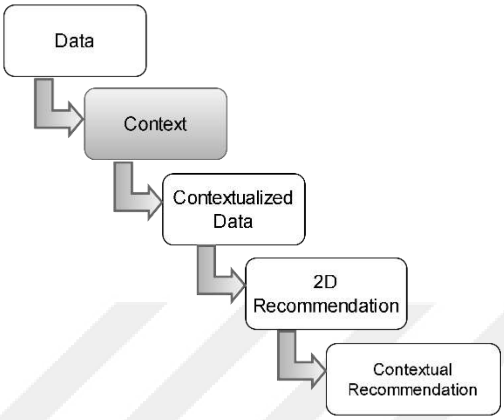

Figure 2.1. *Pre-filtering*

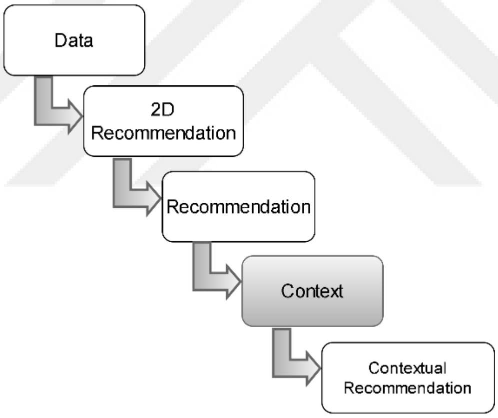

Figure 2.2. *Post-filtering*

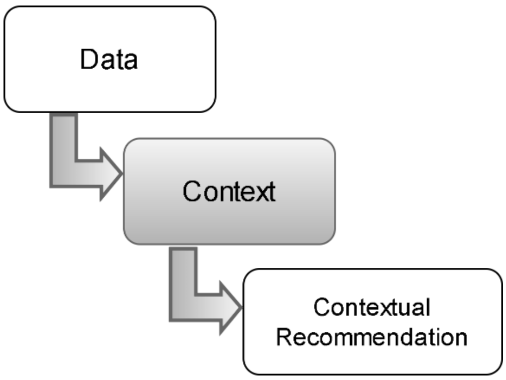

Figure 2.3. *Contextual modelling*

Here we provide a comprehensive overview of the diverse techniques employed in the development of recent CARS based on DL from 2018 to 2024. Additionally, a brief discussion on traditional RS techniques in CARSs also part of this section. The **Figure *2.4*** provides a visual representation of these techniques. The studies reviewed in **Table *2.2*** are categorized based on the different techniques utilized in DL based CARS.

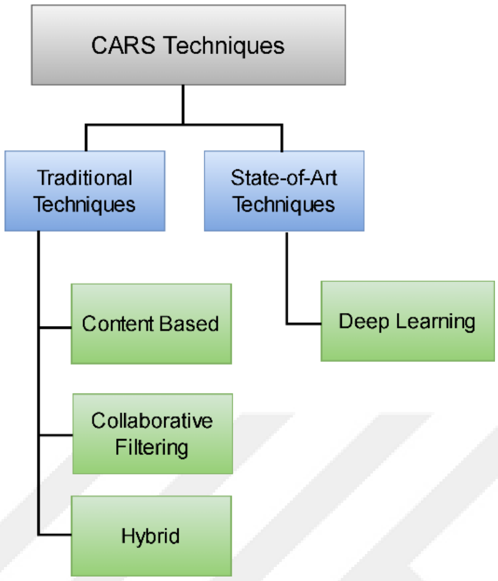

Figure 2.4. *Classification of CARS techniques*

In traditional recommender systems, recommendations are generated based on user preferences, item characteristics, and user-item interactions. CF, CB, and Hybrid approaches are often used in these systems. This section discusses how we can utilize these traditional techniques to combine additional information in the form of context.

The CB is a widely used recommendation approach based on a classic 2D recommender system. The main concept underlying this method is to use the content of each item to make recommendations. It leverages user preferences and item attributes to identify items that are similar to those the user has liked previously. With the flexibility of accepting more information into the recommendation process, contextual data can be integrated into CB.

CB CARS starts with identifying elements in the available data that will define the common dimensional space for describing item characteristics and user preferences. The rating $R(u, i)$ in content-based recommender systems is defined in Eq. (*2.2*).

$$R(u, i) = (user\ profile(u), context(i)) \tag{2.2}$$

u : u represent user profile that represents the user's preferences, behavior, past interactions with the system, and other relevant information that can assist in understanding the user's tastes and interests.

i: represents the context of the item. Context of the items includes features of the item as well as external variables that may impact the user's interaction with the items.

This traditional CARS approach uses various techniques, such as TF-IDF, ontology-based methods, clustering algorithms, Matrix Factorization (MF), probabilistic models, and Tensor Factorization (TF) to build user profiles and suggest relevant items.

Another traditional approach, collaborative filtering is widely used in CARS. The CF method is capable of serving common items to users, as it allows simple communication among people who have the same tastes and preferences. This popular recommendation approach includes primarily two different methods: the memory-based and the model-based (Ricci et al., 2011).

Memory-based CF is only based on user-item interactions. The core mechanism behind this method is to find similarities between users (user-based filtering) or items (item-based filtering) and predict what users might like. In user-based filtering, the systems suggest things those similar users have liked before. And in item-based filtering suggested items that users liked in the past. Distances of these similarities are often calculated by Cosine similarity, Pearson correlation or Euclidean distance (Schafer et al., 2007). Memory-based CF is simple to implement as it does not need to train complex models. Although a limited number of users or items may interact in complex datasets, it has problems with scalability and sparse data.

Model-based CF uses ML algorithm to predict user preferences. Unlike memory-based methods which directly operate on user-item interaction data, model-based techniques are based on algorithms such as MF, Singular Value Decomposition (SVD) and latent factor models to unveil underlying patterns and relationships in the data (Adomavicius & Kwon, 2007). By capturing latent features of items and users, these models are able to predict ratings for items for which users have never interacted before. It can handle large datasets efficiently and provide personalized recommendations with better accuracy. However, model-based CF requires substantial computational resources for training the models and may face cold-start problems (Lika et al., 2014), (Shao & Xie, 2019).

The combination of multiple techniques into hybrid RSs leads to improved recommendation accuracy. Therefore, hybrid approaches overcome the limitations of CF and CB techniques by combining user-item profile data, including both ratings and features.

The use of DL techniques has transformed predictive and analytical models, giving them the ability to act as automatic learners. It makes recommendations more relevant based on context. The main objective of developing DL-based CARS is to improve RS significantly and address the existing issues. The existing literature has presented different DL approaches like Deep Neural Network (DNN), Recurrent Neural Networks (RNN), Convolutional Neural Networks (CNN), and Auto-Encoder (AE).

(Unger et al., 2020) proposed three deep CARS models based on explicit, unstructured, and structured latent representations of various types of contextual data. They empirically shown that these DL-based context-aware models outperformed other ML-based CARS models. Another DL-based model developed by (Liao et al., 2018) integrates multi-context data with a deep neural network model to predict user future locations. This location-based model used both recurrent and feedforward neural networks for identifying sequential behaviors of users with their preferences.

CF is the most widely used traditional RS technique. It is also used in CARS. A CARS-based CF approach is presented by (Jawarneh et al., 2020) that is based on the DL method to incorporate contextual information through a hybrid algorithm. This approach used Neural Collaborative Filtering (NCF) and item-splitting technique to effectively enhance contextualization in recommendations. It also improved recommendation accuracy while handling data sparseness. DeepCARSKit (Zheng, 2022) is an open-source, DL-based recommendation library that includes the implementation of AE, Factorization Machines (FM), and NCF.

CNNs have become a crucial component in the field of RS. Neural Networks with convolution layers and pooling operations are particularly suitable for processing data structured in a grid-like structure. RS can effectively process contextual information using CNNs because they can capture both local and global features efficiently. CNNs have been investigated as a potential development for CARS by Gill et al. (Philosophy, 2022). Recently, another offering of CNN in CARS was introduced by (Hien et al., 2024). They developed a new DL model that combined CNN and MF to improve RSs. This hybrid model combined the benefits of both methods, as a result of which the recommendations became more precise and personalized. The model works well with complex contextual features and shows good results.

In the field of RS, RNNs also have an important composition like CNN. It is best for modeling image data, as they are suitable to work on sequence type. RNNs are different from traditional feed forward neural networks in the sense that it has loops. Those loops have memory of previous computations. Therefore, in the field of RS especially in CARS, the idea of RNN is to model how users past sequential behavior affects user recent action. RNNs with the latent cross (Beutel et al., 2018) used various contextual information inside the RNN model. Latent Cross embedded the context information before and after RNN model hidden states. It was a model formulated in the video domain, taking into account contextual information such as the web page, device, and time of viewing. In (Wu et al., 2022) they applied RNNs and Multilayer perceptron (MLP) for session-based recommendation systems. This work emphasizes utilizing multilayer perceptron to capture the complex dependencies between user, item, and context features. Another context-aware restaurant recommender model using Deep RNNs (Boppana & Sandhya, 2021) achieves 99.6% accuracy.

Auto-encoder is another deep learning technique used in CARS. They are primarily used to reduce the dimensions of input data and to extract their characteristics. (Abinaya & Devi, 2021) proposed a context-specific sentiment analysis-based model along with stacked AEs to enhance effective top-N recommendations. This approach combines user ratings and reviews to address conflicting feedback and creates context-based simulated items. It also tracks user behavior over time as well as conducts sentiment analysis to predict user preference better. Another DL-based context-aware recommendation model presented by (Jeong & Kim, 2022) integrates the AE and NN to extract multiple contextual conditions and predict user preferences.

Recently, a context-aware recommendation model that considers the problem of recommendations as link prediction in a bipartite graph was proposed by (Sattar & Bacciu, 2023) where they present Graph Neural Networks (GNN). The model has incorporated user nodes and item nodes along with context. They particularly focus on the dynamic context influences on the user-item interactions. They also integrate user opinions, static features, and context data into a unified framework. This model significantly outperforms traditional methods by using different types of embeddings.

Moreover, (Jeong & Kim, 2023) suggested a recommender system based on the evolving preference of users. It used a preference transition matrix to capture the changes in preferences of its users and segment data based on time units for temporality.

In this sub-section, we described in detail the work about context-aware recommendation systems. It is now evident that context is crucial in making a user-based recommendation, and the variety of methods helps employ useful types of contexts to generate the desired output for a particular user. In the frameworks of CARS, the types of existing methods include content-based RS, collaborative filtering, hybrid, and deep learning. Although some of the traditional methods can be less effective, deep learning is currently the most appropriate one in terms of dealing with complex data and obtaining users' patterns.

Table 2.2. *CARS using DL techniques*

References	Context	Techniques	Domain
(Hien et al., 2024)	MD	CNN	Movie
(Jeong & Kim, 2023)	MD	NCARS, MF	Cafeteria
(Sattar & Bacciu, 2023)	MD	GNN	Entertainment
(Wu et al., 2022)	Behavior	MLP, RNN	News
(Philosophy, 2022)	MD		Synthetic Data
(Jeong & Kim, 2022)	MD	AE	Entertainment
(Zheng, 2022)		AE, FM, NCF	
(Abinaya & Devi, 2021)	MD	AE	Shopping
(Boppana & Sandhya, 2021)	MD	DRNN	Restaurant
(Unger et al., 2020)	POI	DNN	Restaurant
(Jawarneh et al., 2020)	Time, Location, Trip Type	DL-NCF	Movie, Travel
(Liao et al., 2018)	Location, Time	DNN	Travel
(Beutel et al., 2018)	Time, Device	RNN	YouTube Videos

There are several reasons why traditional CB and CF systems do not necessarily equate to CARS. CB recommendation approaches are limited by their inherent static nature and lack of contextual understanding. These systems are not designed to accommodate the dynamic and context-specific preferences of users, resulting in less effective context-aware recommendation systems. For instance, a user with a preference for action movies may have a content-based recommender suggest more titles within that genre. However, if the user is currently seeking a family film to watch with children or wife, the system would still recommend action movies, which may not be suitable in that particular context. Furthermore, traditional recommendation systems like CF often struggle to adapt to the dynamic and context-specific preferences of users. For example, when a user is searching for restaurant options, a traditional CF system might suggest restaurants similar to ones the user has liked previously. However, if the user is traveling with friend and prefers local food, or reached late at night and requires 24-hour options, these recommendations may not be suitable. In contrast, a CARS would consider the current situation and suggest more appropriate suggestions.

While traditional hybrid recommendation techniques integrate the benefits of CF and CB to enhance accuracy, they fall short in adapting to dynamic and context-specific user preferences. Without the ability to incorporate real-time contextual information, these systems cannot fully address the diverse and evolving needs of users in different situations. Like in the previous example, a user searching for restaurant recommendations while traveling to a new city may prefer local food, but a traditional hybrid system may still suggest old city restaurants that are similar to those in the user's home location, disregarding the relevant travel context.

CARS can benefit from DL techniques by dynamically adapting to changing contexts, handling high-dimensional and complex data, automatically learning relevant features, learning non-linear relationships, scaling well, and incorporating diverse data sources. Consequently, DL delivers highly personalized and contextually relevant recommendations, surpassing traditional methods such as CB, CF, and Hybrid methods.

2.2. Multi-Criteria Recommender System

Recommender systems are commonly assessed based on a single metric, such as the overall rating a user assigns to an item. However, this approach is limited, as users typically consider various factors when selecting an item (Adomavicius & Tuzhilin, 2005b), (Manouselis & Costopoulou, 2007), (Adomavicius & Kwon, 2007). Multi-criteria recommender systems seek to address this limitation by predicting item ratings across multiple criteria, offering more detailed insights into user preferences (Bilge & Kaleli, 2014). These multiple criteria ratings represent various and important preferences of users. This makes MCRSs more informative and useful since they make accurate recommendations.

The Conventional recommendation systems employ a two-dimensional matrix to model the relationship between users and items, where a user's overall rating of an item is denoted in Eq.(*2.3*).

$$R: Users \times Items \rightarrow R_0 \tag{2.3}$$

These systems primarily rely on the overall rating when making recommendations to users, but the information provided by overall ratings is often inadequate to capture the specific reasons underlying user preferences. In contrast, MCRSs aim to predict an item's rating across multiple factors or criteria, which can provide more meaningful insights. The various criteria ratings represent distinct and essential aspects of user preferences. Consequently, MCRSs appear more informative and valuable as they make accurate recommendations while considering multiple user preference dimensions, ultimately leading to enhanced recommender system performance (Bilge & Kaleli, 2014). While some multi-criteria recommendation systems may omit the overall rating and concentrate solely on individual criterion ratings, the rating function in a MCRS can be formulated in Eq. (*2.4*) and Eq. (*2.5*), irrespective of whether an overall rating is present or not.

$$R: User \times Item \rightarrow R_0 \times R_1 \times R_2 \times \ldots . \times R_c \tag{2.4}$$

Or

$$R: User \times Item \rightarrow R_1 \times R_2 \times R_3 \times \ldots . \times R_n \tag{2.5}$$

where:

R_0 : represents the overall rating.

R_1 to R_c : represent the MC ratings.

Incorporating multi-criteria user ratings can enhance the efficacy of recommendation systems by reflecting the nuanced nature of user preferences. An example of a multi-criteria rating is presented in **Table *2.3***.

Table 2.3. *Representation of multi-criteria ratings*

User	Ratings
User $_1$	5 $_{(2,2,8,8)}$
User $_2$	5 $_{(8,8,2,2)}$
User $_3$	6 $_{(3,3,9,9)}$
User $_4$	7 $_{(5,5,9,9)}$
User $_5$	4 $_{(8,8,2,1)}$

Let Suppose **Table *2.3*** contains user-item ratings data for a restaurant dataset, where users rate restaurants based on various criteria like service, location, cleanliness, and overall rating. These multi-criteria ratings provide a more comprehensive understanding of user preferences, leading to more accurate recommendations. For instance, if predictions are made for User 2, the overall rating may suggest similarity to User 1. However, when considering multi-criteria ratings, User 2 is more similar to User 5. This shows how the overall rating can obscure users' underlying preferences for different aspects of a particular item. Moreover, ratings based on multiple criteria can provide more targeted and practical recommendations. Therefore, in the field of recommender systems, the importance of MCRS has been acknowledged as a distinct area of research and development (Adomavicius & Tuzhilin, 2005b), (Adomavicius et al., 2011), (Manouselis & Costopoulou, 2007). Various web-based applications, including platforms such as Buy.com, Yahoo! Movies, and Zagat's Guide, employ MCRS to offer users efficient and reliable recommendations (Adomavicius et al., 2011).

Multi-criteria ratings have recently received great attention and found real-life applications in the next generation of recommendation systems. An increasing number of recommender systems have integrated their system with multi-criteria ratings to improve the quality of recommendations produced by these systems (Zheng et al., 2019). Therefore, a variety of recommendation techniques and algorithms targeted at recommender systems that may benefit from multiple ratings have been developed. As MCRSs have been developed as an extension of CF-based RSs, recommendation techniques based on multi-criteria ratings can be divided into two types: heuristic-based, also called memory-based and model-based approaches, all of which intend to predict and recommend either an overall rating or individual criterion ratings. (Adomavicius et al., 2011). The similarity in memory-based methods for multiple criteria can be calculated in two different ways (Adomavicius & Kwon, 2007). The first approach uses a standard method to compute the similarity between each criterion rating individually and then uses an aggregation method, such as the average, worst-case, which aggregates the values. And, the second way calculates the distance between multi-criteria ratings using multidimensional distance metrics including (Euclidean, Manhattan, and Chebyshev distance metrics). On the other hand, statistical or machine-learning methods are frequently used by model-based approaches to construct predictive models. The learned model is then applied to predict the utility of unknown items for recommendation purposes.

Various significant algorithms have been widely used in different research studies in MCRSs based on memory and model-based techniques (Monti et al., 2021), (Wasid & Ali, 2018). A brief literature review of MCRS techniques is also a present of this chapter, which is divided into two sub-parts: MCRS using machine learning techniques and MCRS using deep learning techniques. **Table *2.4*** lists studies using different techniques in MCRS.

Table 2.4. *MCRS techniques*

Techniques	References
ML	(Gupta and Kant 2020), (Esteban, Zafra, and Romero 2020), (Nilashi et al. 2019), (Zheng 2019b), (Wasid and Ali 2018), (Kant, Jhalani, and Dwivedi 2018), (Zheng 2017), (Ranjbar Kermany and Alizadeh 2017), (Shambour, Hourani, and Fraihat 2016), (Jhalani, Kant, and Dwivedi 2016)
DL	(BATMAZ and KALELİ 2022), (Shambour 2021), (Nassar, Jafar, and Rahhal 2020b), (Nassar, Jafar, and Rahhal 2020a), (Batmaz and Kaleli 2019a), (Tallapally et al. 2018), (Hassan and Hamada 2017)

Various studies that have been listed in **Table *2.5***, different ML techniques, such as CF methods, regression techniques, clustering algorithms, and classification methods, in order to increase the performance of MCRSs. These techniques have allowed researchers to develop more sophisticated recommender systems. They may capture better user preferences as well as understand complex relationships between items, and provide more personalized suggestions to users.

ML-based MCRS study proposed by Kant et al. (Kant et al., 2018) leverages a Fuzzy Naive Bayesian (FNB) approach to address the uncertainties inherent in user preferences and the correlation-based similarity challenges encountered in traditional collaborative filtering methods. Another, (Jhalani et al., 2016) presents a MCRS that uses multi–Linear Regression (LR) approach to determine the weights for the different criteria and aggregate the similarities between ratings. Further, an Item-based Multi-Criteria Collaborative Filtering (IMCCF) algorithm for personalized recommendation systems is proposed by (Shambour et al., 2016). The IMCCF method leverages both the semantic information of items and the multi-criteria ratings provided by users. It overcomes the limitations of traditional item-based CF techniques, such as data sparsity.

In MCRS, Genetic Programming (GP) is also used to learn weights for each criterion. (Gupta & Kant, 2020) proposed a genetic programming-based multi-criteria RS. Another ML-based MCRS approach is presented in (Wasid & Ali, 2018), they calculated similarities between users based on different criteria by using Euclidean distance. After that, results are used for k-means clustering to generate overall recommendations. Typically, MCRS approaches predict ratings independently for each criterion, and then aggregate them to predict user preferences. In contrast, the criteria chains approach (Zheng, 2017) considers the interdependencies between the various criteria by predicting the ratings for each criterion sequentially. It uses the previous predictions as contextual information.

Some hybrid approaches based on ML are also used in MCRS. A hybrid item-based ontological semantic filtering technique was proposed by (Ranjbar Kermany & Alizadeh, 2017). They explore the complex relationship between multi-criteria ratings and overall ratings using demographic information. Another hybrid approach was introduced by (Nilashi et al., 2019) to provide recommendations for travelers. This technique combines dimensionality reduction through Higher-Order Singular Value Decomposition (HOSVD) and predictive modeling using Adaptive Neuro-Fuzzy Inference Systems (ANFIS). The HOSVD is employed to decompose the high-dimensional hotel rating data and identify similarities among users, while the ANFIS is used to predict the overall ratings. The hybrid approach (Esteban et al., 2020) for MCRS integrates CF and CB filtering for student data. In this study, authors use a genetic algorithm to optimize the weights of different criteria.

(Zheng, 2019b) proposed a utility-based multi-criteria recommendation algorithm where user preferences were learnt with the help of different learning-to-rank methods. He introduces the vector space model approach that describes similarity or distance between a multi-criteria rating. The author also compares their utility-based models with several baseline approaches such as matrix factorization, linear aggregation, hybrid context, criteria chain and flexible mixture model.

Table 2.5. *MCRS using ML techniques*

References	Techniques	Domain
(Gupta & Kant, 2020)	GP-MCRS	Movie
(Esteban et al., 2020)	Hybrid	Education
(Nilashi et al., 2019)	HOSVD, ANFIS	Tourism
(Zheng, 2019b)	Utility-based MCRS	Tourism, Movie
(Wasid & Ali, 2018)	Clustering	Movie
(Kant et al., 2018)	FNB	Movie
(Zheng, 2017)	Multiple methods	Movies, Tourism
(Ranjbar Kermany & Alizadeh, 2017)	Ontology	Movie
(Shambour et al., 2016).	IMCCF	Movie
(Jhalani et al., 2016)	LR	Movie

DL algorithms are capable of working with a large and complicated set of data as well as complex non-linear interrelationships between multiple criteria. However, ML algorithms like LR or MF often assume linearity and sometimes explicitly model interactions which does not scale well. Hence, DL techniques, like DNNs, RNNs and AE are able to realize superior performance in MCRS against ML techniques. Moreover, they have the advanced capability to model intricate relations that are non-linear between many factors. By applying DL techniques, MCRS can offer a more accurate and vast span of recommendations to the users.

In MCRS, DNN is the most frequently used deep learning technique. A DNN-based model is presented in (Hassan & Hamada, 2017). It enhances the predictive performance of MCRS. This model is trained using simulated annealing algorithms and then integrated it with single-rating recommender systems. A novel deep MC-CF model for recommendation systems is proposed by (Nassar et al., 2020a). This model also used the DNNs framework to predict the multi-criteria ratings and learns the aggregation function between the multi-criteria ratings. Another study, (Nassar et al., 2020b) presents an MC-CF recommendation system that integrates DNN and MF. The proposed model consists of two components, the first one employs an integrated DNN and MF approach to predict the multiple criteria ratings, while the second one uses a DNN to predict the overall rating.

Another DL technique is auto-encoders that enhance MCRS by reducing dimensionality and improving prediction accuracy. An AE architecture is designed by (Tallapally et al., 2018) to address the limitations of traditional RSs. MC ratings are used as inputs to this model, which uses it to learn how to give better recommendations based on the relationship between the criteria ratings and the overall ratings. Another MCRS approach using the AE technique is presented in (Batmaz & Kaleli, 2019). This model predicts user ratings for each criterion independently, and a multilayer perceptron model is used to estimate the overall rating.(BATMAZ & KALELİ, 2022) also proposed a new similarity-based MC-CF algorithm called AE-sim MCCF that leverages AE to address the sparsity issue in MCCF systems. The experimental findings of the proposed model demonstrate that it can better overcome the negative effects of sparsity compared to other baseline MCCF methods, in terms of accuracy and coverage. Another study presented in (Shambour, 2021), is also based on AE recommendation algorithm. This algorithm is used to capture the complex, nonlinear relationships between users and items. **Table *2.6*** shows MCRS that uses deep learning techniques.

Table 2.6. *MCRS using DL techniques*

References	Techniques	Domain
(BATMAZ & KALELİ, 2022)	AE-sim MCCF	Movie
(Shambour, 2021)	AEMC	Movie, Tourism
(Nassar et al., 2020b)	DNN & MF	Tourism
(Nassar et al., 2020a)	DNN	Tourism
(Batmaz & Kaleli, 2019)	Auto-encoder MCCF	Movie
(Tallapally et al., 2018)	Auto-encoder	Movie, Tourism
(Hassan & Hamada, 2017)	NN	Tourism, Movie

In this sub-section, we provided a detailed overview of MCRSs. We conclude that MCRS emphasizes the important role of supplementary information in generating personalized recommendations. We also examined the capabilities of ML and DL approaches for MCRSs. These techniques are designed to deliver tailored suggestions that align with users' individual preferences.

MCRS that uses DL algorithms especially DNN have various benefits over traditional ML algorithms. These benefits comprise the ability to seamlessly handle massive amounts and variations of data, pattern discovery automatically as well representing sophisticated non-linear correlations among multiple factors. In contrast, many ML models such as LR or MF often make linearity assumptions and need to explicitly model interactions which can be unwieldy and less optimal. Deep learning algorithms also support various data sources and can scale with ease to distributed training on large-scale datasets, fully understanding the end-to-end recommendation process. We know that ML algorithms usually have complex coordination of different models for different data types. Further, does not scale well to large datasets and needs a lot of computation and experience with optimization tricks as well.

3. LITERATURE ON CONTEXT-AWARE MULTI-CRITERIA RECOMMENDER SYSTEM

CARS and MCRS are shown to outperform traditional RSs for a variety of scenarios (Odić et al., 2012). Although the CARS and MCRS approaches have made significant progress, they still need to be improved further to allow seamless implementation in a wide range of real-time applications. The combination of these two approaches has become a focus of recent research to improve the accuracy of results. The integration of CARS and MCRS has already been investigated in 13 studies. However, there is no literature review on Context-aware Multicriteria recommendation systems.

This chapter aims to argue the significance of CMRS in predicting and improving RS performance by illustrating the efficiency gained from integrating both CARS and MCRS that would offer a step further for enhancing accuracy. This chapter addresses CMRS studies in the time range 2012-2023 that explore context, criteria and integration techniques, datasets as well as evaluation metrics.

Criteria have been used as context in some studies of CMRS. Like, a context-aware framework named JHPeer (Yeung et al., 2012) is intended to facilitate the development of context-aware applications in mobile settings. This employs a peer-to-peer network architecture and offers components for managing context, including sharing and distributing context information across disparate applications. Additionally, the framework enables the proactive delivery of information to users. The paper proposes an AHP-MCR recommendation approach that incorporates additional contextual factors (criteria) into the ranking process using the Analytic Hierarchy Process (AHP). The implementation of a proactive, personalized news recommendation application is presented to demonstrate the viability of the proposed framework and approach. Another study (Zheng, 2019a) presented criteria as context to enhance the effectiveness of MCRS. Evaluations of three real-world datasets reveal that the methods using criteria preferences as context generally outperform traditional MCRS techniques.

Some studies have not explicitly defined criteria. They used context as the criteria. (Nguyen & Nguyen, 2015) developed a CARS that leverages multi-criteria communities. This system employs two algorithms, Contextual Rating Multi-Criteria Communities (CRMC) and Contextual Rating Enriched by Similar Communities (CRESC), to generate personalized recommendations based on users' dynamic contexts. CRMC identifies the most relevant community, while CRESC enhances the selected community with similar communities to improve recommendation quality. (Ashley-Dejo et al., 2016) proposed a context-aware active recommendation system for tourists that used location, time, and weather data as context to recommend relevant destinations. The authors used a hybrid MC-CF approach to provide personalized recommendations based on user ratings across multiple contexts. The system architecture consists of a client-side and a server-side, where the client transmits contextual information to the server, which then generates user profiles to deliver personalized recommendations.

A few CMRS studies have not used real-world datasets for their experiments. (Rodríguez & Viktor, 2013) a study introduced a personalized, location-aware multi-criteria recommender system that constructs user preference models by considering various factors, such as user demographics, item attributes, distance, and travel time. The system first establishes a user profile by determining the demographic clusters the user is associated with and the weights the user assigns to different item attributes. It then evaluates the utility of each recommended item based on the user's location, mobility level, and the established preference model. The performance of the system was assessed through an offline experiment involving 30 participants. The results show that the system can accurately identify the demographic groups a user belongs to and generate accurate recommendations that align with the user's preferences. Another food recommendation system presented by (Toledo et al., 2019) considers both nutritional information and user preferences. This system uses multi-criteria decision analysis techniques such as the AHP and AHPSort to evaluate a food's nutritional suitability based on factors like protein, sodium, cholesterol, and saturated fats. It also takes user preferences into the recommendation process and uses an optimization-based method to create menus. The paper evaluates this approach through a case study.

A number of CMRS studies omitted empirical investigations. Like in (Gasparic & Ricci, 2015), they presented an advanced command prediction system that takes into account various criteria and contextual elements in order to improve the quality of recommendations. With this model, users' preferences and diverse contextual factors are combined to provide more relevant suggestions to software developers. This model is based on the Unified Theory of Acceptance and Use of Technology (UTAUT). It incorporates performance expectancy, effort expectancy, and social influence to anticipate a developer’s willingness to utilize a suggested command. A developer's acceptance probability is evaluated based on these characteristics, which are considered distinct criteria within the model. The authors consider four types of contexts: current practices, environments, interactions, and the presentation of recommendations. The study lacks experimental findings or results. (Ramchandani et al., 2020) presented a CMRS designed for farmers. The system provides personalized recommendations based on multi-criteria like soil fertility, crop type, soil composition, and weather conditions. The study aims to boost productivity by making recommendations tailored to farmers' specific needs. The system uses a weighted method to evaluate crop suitability and uses cloud computing to make the recommendations ubiquitously accessible to farmers. However, the study does not include empirical findings.

Most CMRS studies have been based on ML algorithms. A context-aware and multi-criteria decision-making framework in the educational field (Zheng et al., 2019). It explores several machine learning-based approaches to incorporate contextual information into the processes of predicting and integrating multi-criteria ratings. The findings indicate that the integration of CARS and MCRS can enhance recommendation performance compared to using either approach individually. A multi-criteria context-aware music recommendation approach is presented in (Sassi et al., 2021). This approach addresses problems in the music recommendation process, such as playlist diversity, scalability, and cold start. It evaluates the relevance of contextual information and uses a weighted aggregation technique to combine user ratings. To recommend matching genres, and generates diverse playlists it employs a clustering-based model. Experiments show the approach outperforms traditional and context-aware baselines in precision, recall, and F-measure. (Dridi et al., 2019) introduces a novel contextual multi-rating system approach that simultaneously clusters users' situational contexts and the criteria by which they evaluate items. The authors maintain that their strategy takes advantage of both contextual information and multi-dimensional ratings to generate more precise predictions. For the task of rating prediction, the paper employs matrix factorization as the algorithm operating on the sub-matrices obtained from the co-clustering step. While matrix factorization is utilized for the multi-criteria rating prediction, the specific algorithm used for the context-aware component is not specified. Another contribution in CMRS by Dirdi et al. (Dridi et al., 2022). They proposed an ML-based model that leverages the relationships between users, contextual situations, and item criteria to provide personalized recommendations. Model completion involves the following key steps: (1) modeling the multi-dimensional data as a tripartite graph, (2) performing high-order co-clustering on the graph to partition users in similar contexts and their evaluated criteria, (3) introducing an improved rating prediction method that considers user-context dependency and criterion correlation, and (4) aggregating the predicted multi-criteria ratings into a single overall item rating. Evaluation of real-world datasets demonstrates the model's superior recommendation performance compared to baseline approaches of CMRS.

A DL-based CMRS presented in (Vu & Le, 2023), This study used two DNNs. The first model predicts criteria ratings and the second model is used for overall ratings. A key contribution of this approach is leverages DL techniques to enhance accuracy and personalization in context-aware MCRS. The authors used a contextual pre-filtering method to predict criteria ratings and evaluate them individually with the DNN model. Another DNN model is used to predict the overall rating, utilizing the output from the initial neural network as its input. The results show that contextual pre-filtering and the integrated DNN model outperform standalone DNN models in MCRS.

We investigated various integrated techniques for CARS and MCRS in this section. Existing studies on integrating CARS and MCRS have primarily utilized traditional ML techniques. However, these existing works have certain limitations, such as focusing on a single context (e.g., only considering user preferences or item features), using context as criteria or criteria as context without a clear distinction, lacking thorough experimental validation to assess their performance, and relying solely on conventional ML algorithms that may not fully capture the complex relationships between various contexts factors in CARS and MCRS, as shown in **Table *3.1***.

Table 3.1. *CMRS studies with their limitations*

References	Context	Criteria	Techniques: ML	Techniques: DL	Limitations
(Yeung et al., 2012)	✗	✓	✓		Criteria were based on context. There is no separate context available.
(Rodríguez & Viktor, 2013)	✓	✓	✓		A pilot study was conducted for the experiment instead of a real-time dataset.
(Gasparic & Ricci, 2015)	✓	✓	✓		No empirical study was performed.
(Nguyen & Nguyen, 2015)	✓	✗	✓		Absence of Multi-criteria rating.
(Ashley-Dejo et al., 2016)	✓	✗	✓		Criteria are not mentioned.
(Toledo et al., 2019)	✓	✓	✓		The research was conducted through a case study-based experiment, rather than a real-time dataset.
(Dridi et al., 2019)	✗	✓	✓		There is no defined approach for CARS.
(Zheng et al., 2019)	✓	✓	✓		Researchers used traditional ML algorithms.
(Zheng, 2019a)	✗	✓	✓		This study employed a single approach, as it used criteria as the context for its analysis.
(Ramchandani et al., 2020)	✓	✓	✓		No experimental results were presented.

(Sassi et al., 2021)	✓	✓	✓		The study is based on traditional ML algorithms.
(Dridi et al., 2022)	✓	✓	✓		This research also used machine-learning techniques.
(Vu & Le, 2023	✓	✓		✓	This study utilized a single-context, which may not be the most optimal choice for analyzing a multidimensional dataset in DL.

4. AN APPROACH FOR MULTI-CONTEXT MULTI-CRITERIA RECOMMENDER SYSTEM BASED ON DEEP LEARNING

This section presents the architecture of our proposed recommender system. It consists of two distinct but interconnected components. First, we carefully extract contextual conditions and multicriteria item ratings and feed them into a DNN-based composite model. In the second phase, a sophisticated DNN-based MCoMCRS is used. Through the processing and synthesis of multiple contextual factors and criteria-based evaluations, this model predicts overall item ratings.

By considering the context in which the recommendation is made and multiple criteria for evaluating items, this approach utilizes the strengths of both CARS and MCRS. To our knowledge, our research is the first to apply multi-contextual data within a DNN-based MCRS.

Furthermore, we observed that our proposed model leads to significant improvements over other state-of-the-art methods based on experiments on real-world Trip advisor and ITM-Rec datasets that will provide in chapter 6.

The Proposed Algorithm

In our study, we develop an advanced Multi-Context Multi-Criteria (MCoMC) algorithm based on deep learning. This algorithm is characterized by a sophisticated neural network architecture that has been carefully designed to integrate numerous contextual variables with a multidimensional set of criteria. The primary objective of our study is to develop a detailed and accurate projection of anticipated user preferences. The model we have developed employs a multi-faceted rating system alongside various contextual factors, which can be combination of Eq. (*4.1*) and Eq. (*4.2*).

$$R : User \times Items \rightarrow Cr_1 \times Cr_2 \times Cr_3 \times \ldots . \times C_k \tag{4.1}$$

$$R : User \times Items \rightarrow Co_1 \times Co_2 \times Co_3 \times \ldots . \times Co_k \tag{4.2}$$

In the first equation, we model a MCRS where function R captures the relationship between user-item interactions and multiple criteria ratings Cr_1 to Cr_k. And, the second equation represents a CARS, where user-item interactions are mapped to various contextual factors Co_1 to Co_k.

By integrating the Eq. (*4.1*) and Eq. (*4.2*), the resulting proposed formulation is presented in Eq. (*4.3*).

$$R : co \times cr \times x \times y \rightarrow Rating \tag{4.3}$$

R signifies the predictive rating function, targeting an item y for a user x, factoring in criteria cr within a specific context co. This integrated function offers a prediction across various conditions. It utilizes the strengths of both CARS and MCRS by analyzing the context of the recommendation and evaluating items based on multiple criteria.

The development of a deep learning-based MCoMC algorithm involves a multistage process to integrate ratings derived from multiple contexts and criteria. This process comprises four steps i) dataset preparation ii) model architecture design and training iii) loss function formulation and iv) model evaluation. These preprocessing steps are essential for preparing the data to maximize the effectiveness of the subsequent modelling steps, ensuring that our model can learn from well-structured and consistent input data. The detailed description of pseudo-code for the algorithm is delineated in below.

This section provides a detailed analysis of each individual process of the proposed algorithm.

Algorithm: The proposed MCoMC algorithm
Input: dataset $\boldsymbol{R}_{co \times cr \times x \times y}$ ← *co: context, c: rating criteria, x: user, y: item* **Output** ← *overall prediction of ratings* **Phase 1: Multi-context, Multi-criteria Dataset Preparation** **1)** Extract the number of criteria, context, users, and items from $\boldsymbol{R}_{co \times cr \times x \times y}$ **2)** For each criterion, context Do: **a)** Fill missing values with **Mean** **b)** Split dataset into the random train, validation, and test sets **Phase 2: Deep Neural Network Construction** **1)** Define input and output layer sizes based on feature dimensions **2)** Define the architecture of hidden layers (e.g., number of layers, neurons per layer) **3)** Initialize weights (consider advanced initialization techniques) **4)** Define activation function: **ReLU** **5)** Define regularization (**Dropout = 0.2**, explore other regularization methods) **6)** Define optimizer: **Adam** **7)** Define loss function: **MAE, RMSE** **8)** Define the number of epochs **Phase 3: Training Validation and Testing** **1)** For each epoch Do: **a)** Input contexts, criteria, user, and item features into the neural network **b)** Forward pass through the layers **c)** Calculate loss **d)** Update weights and biases using backpropagation and Adam optimizer **2)** Compute the overall rating for the test set using the trained model **3)** Calculate test set loss **4)** Report model performance

Step 1: Multi-context, Multi-criteria Dataset Preparation

In order to construct a dataset suitable for our deep learning model, we devised a methodical approach utilizing Python to extract and preprocess data encompassing multiple criteria and multiple contexts. This technique is designed to be adaptable to any dataset enhanced with diverse user-generated content, with a focus on ensuring the algorithm's resilience across various application domains. The preparatory process entails several pivotal steps to optimize the data for our model:

i) Data Extraction

We extract and analyze a wide set of relevant features indicating contextual information and multiple evaluative criteria to provide a comprehensive assessment. The result of this analysis provides us with a comprehensive dataset in the form of multiple contexts, multiple rating criteria, and overall ratings. The contextual situations include time, location, and companion that influence user preferences. Whereas criteria capture multiple aspects of rating. For example, quality, service, and value, etc. The user includes individual identifiers for each user, while the item dimension includes unique identifiers for each rated item. We identified at least four contextual conditions to capture user preferences within the datasets. This results in a robust and multi-faceted dataset that enables us to gain deeper insights into the factors, shaping user experiences and behavior.

ii) Handling Missing Values

It is essential to deal with missing values before entering data into the DNN model. This process ensures data quality and integrity. It also improves the performance and reliability of the model. Unhandled missing values can change the statistical properties of the dataset, leading to biased findings and inadequate model performance. To mitigate the effects of missing data, we employed Mean Imputation. It utilizes all available data points, maintains dataset size, and preserves the overall data distribution. When a user omits to provide a rating for a specific item, the dataset automatically substitutes a mean value, maintaining continuity and uniformity in the interpretation of the data. This simple and efficient method helps the DNN to learn meaningful patterns without disrupting the learning process.

iii) Data Encoding

To capture and leverage contextual information, data encoding is a crucial aspect of DNNs. This encryption leads to improved model performance, enhanced generalization capabilities, and a strengthened ability to uncover complex relationships within the data. We used One-hot encoding technique that converts categorical data into a numerical format for our model. Our model treats each contextual condition as a binary vector with a single '1' element, making it able to process contextual data without assuming ordinal relationships. The categorical nature of the data is preserved by this technique and each category is treated as a unique feature. Resultantly, this enhances the model's ability to learn from context and improves accuracy.

iv) ***Feature Scaling***

Normalizing the data ensures that all features receive equal attention during the training process. It also accelerates the model's convergence and enhances its overall effectiveness. So, we employ Min-Max scaling across all features to transform the data into a uniform scale. This crucial step prevents the dominance of features due to their original range, which could affect the learning process of the model.

The step-by-step process underlying our model, ranging from the preliminary stage of data acquisition to the final stage of generating predictive outputs, is systematically depicted in **Figure *4.1***.

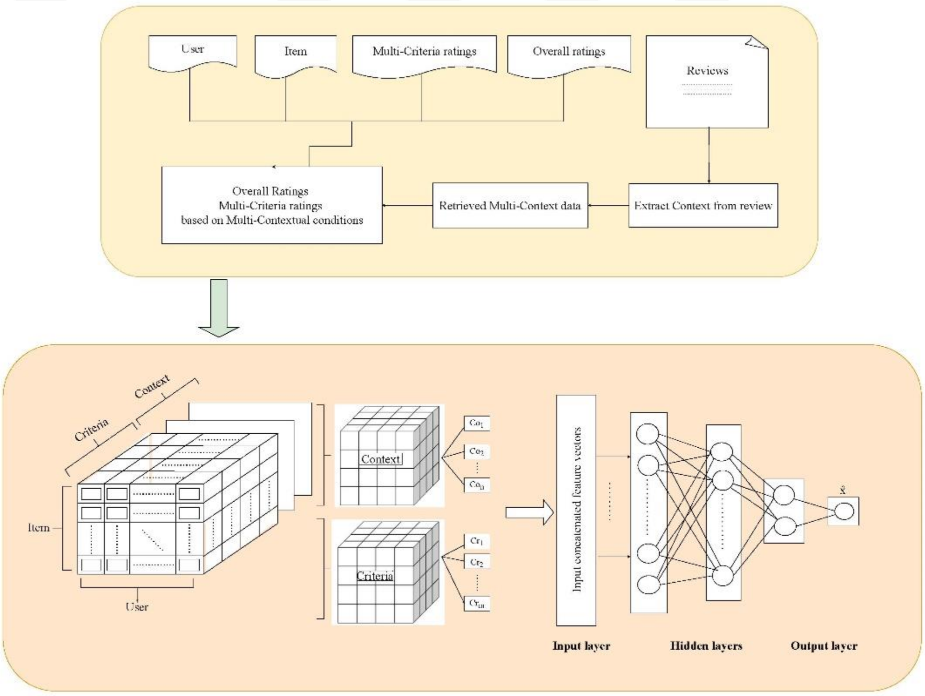

Figure 4.1. *Operational framework of the proposed model*

The initial stage of our research framework basis upon the careful compilation of datasets. This stage captures contextual factors and multiple criteria, all of which are subject to user ratings. This preparatory phase requires the careful construction of a multidimensional data structure, drawing from the Trip Advisor (Batmaz & Kaleli, 2019) and ITM-Rec (Zheng, 2023). These datasets contain several contextual variables along with their overall ratings. We have integrated this multi-context, multi-criteria rating approach as a fundamental component throughout the development of our deep learning model.

Step 2: Deep Neural Network Construction

After the preparation of the dataset, we developed a DNN-based multi-criteria and multi-context recommendation system. This network is not just a set of layers, but a carefully designed system of interconnected nodes. Each node plays an important role in processing and interpreting complex data patterns. The network architecture includes an input layer that serves as the entry point for the multi-criteria and contextual data. After this, multiple hidden layers allow the network to discover and learn complex data relationships. These layers are a deliberate choice based on the complexity of the dataset and the problem at hand. For our particular application, we have chosen three hidden layers. As an activation function, we choose the Rectified Linear Units (ReLU), described in Eq. (*4.4*).

$$ReLU\ (z)\ =\ max(0, z) \tag{4.4}$$

where (z) represents any hidden layer's input. The utilization of the ReLU activation function is a purposeful choice because of its proven efficacy in introducing non-linearity. By doing so, the network can model complex functions more effectively.

To strengthen the network's resilience and improve its generalization capacity, a dropout regularization strategy with a 0.2 dropout rate is implemented, thus preventing overfitting. Moreover, to optimize the model's performance Adam optimizer is integrated into the network, recognized for its efficiency and robustness to noisy data.

Step 3: Loss Function

Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and other loss functions are critical in stimulating the training activities of neural networks. The network tries to minimize these, during its training phases. Continuous adjustment of the network's internal parameters in the training process improves the performance of the model and allows it eventually to converge upon the best possible setting. For target accuracy to its desired standard, these loss functions are vital to the network's learning process.

MAE and RMSE are excellent statistical measures that provide respective rationale to a predictive model in general. Therefore, we included these metrics in our evaluation.

Step 4: Neural Network Training and Testing

The final stage of the model is subjected to extensive training and rigorous testing protocols utilizing the previously compiled dataset. The dataset is purposefully divided into distinct segments, with 70% allocated for training purposes, 10% for validation, and the remaining 20% for testing purposes.

There are two main objectives of this phase 1) to optimize the neural network parameters for minimal loss function values and ii) to enhance the model's predictive performance.

Empirical investigation revealed that 150 training iterations represented the optimal number. Following the completion of the training process, the model's predictions integrate multiple assessment criteria and diverse contextual factors to generate an aggregated prediction for each item, as depicted in **Figure *4.1***.

Step 5. Model Evaluation

As a final step, the model is thoroughly evaluated, juxtaposed against other baseline models to determine its performance. Ascertaining the efficacy and superiority of our proposed MCoMCRS algorithm over existing algorithms requires this comparative analysis.

The proposed MCoMCRS integrates two promising techniques of RS to provide more personalized and accurate recommendations. This DNN model captured the users, items, contexts, and multiple criteria ratings to generate recommendations. The model enables the assessment of user preferences across a wide spectrum of dimensions, which can enhance personalization, and improve the accuracy of evaluations. This model is particularly well-suited for applications in which user preferences exhibit a multidimensional as well as influenced by contextual conditions.

5. EXPERIMENTAL EVALUATION

The proposed technique, which uses DNN, multi-contextual data, multi criteria ratings, must be compared to the existing methods, which use DNN on MCRS and CARS.

In this section we described a comprehensive evaluation of the proposed MCoMCRS method, using two publicly available datasets that are discussed in the following sections.

5.1. Datasets and Evaluation Metrics

n this section, we describe the datasets that were used for evaluating the proposed MCoMCRS model. We also discussed the context and the scale used in these datasets. The datasets showed that the model handled multi-criteria and multi-context data effectively in real-world settings.

TripAdvisor and ITM-Rec datasets are used in experiments. The detailed description of these datasets is provided in **Table *5.1***. Based on reviews from 3,494 users, the TripAdvisor dataset contains 3,708 ratings across 485 hotels, sourced from(Batmaz & Kaleli, 2019). Specifically, the datasets contain four contextual dimensions: trip companion, trip reason, meal, and season (for example, trip companion includes nine conditions, such as family, friend, etc.). A further six criteria are incorporated into the dataset, namely location, cleanliness, value, rooms, service, and sleep quality, with a rating scale of 1 to 5.

From 2017 to 2022, the ITM-Rec dataset (Zheng, 2023) includes 5,230 ratings from 454 users for 70 students. Three contextual dimensions are included: Courses (Database (DB)/Data Science (DS)/Data Analytics (DA)/Semester (Spring and Fall) and the lockdown periods during COVID-19 (PRE, DUR, POS). In addition to the overall score, three criteria ratings are considered: App, Data, and Ease.

Table 5.1. *Dataset description*

	TripAdvisor	**ITM-Rec**
No. of Users	3494	454
No. of Items	485	70
No. of Ratings	3708	5230
Rating Scale	1-5	1-5
Contextual Dimensions	4	3
Contextual Conditions	33	8
Data Sparsity	99.78%	83.54%

The extensive collection and multifaceted nature of these data sources underscore the reliability and rigorous nature of our experimental results, further demonstrating the capability of our model to effectively manage complex, large-scale data environments.

For a comprehensive evaluation of the model's performance across different configurations, we divided each dataset into training, validation, and testing segments using a (70/10/20) ratio respectively. It was necessary to include one-hot encoding of categorical variables and normalization of continuous variables to ensure uniform input scales for the neural network as a preprocessing step.

For our evaluation, we selected MAE and RMSE as the primary metrics, in alignment with the accepted standards in the field of rating prediction studies (Shani & Gunawardana, 2011). MAE assesses bias within ratings by quantifying the absolute deviation between predicted and actual ratings. In contrast, RMSE emphasizes the impact of significant errors. It represents the standard deviation between the predicted and actual rating values. These metrics are defined in below Eq. (*5.1*) and Eq. (*5.2*), respectively,

$$MAE = \frac{\sum_{i=1}^{n}(\widehat{x_i} - x_i)^2}{n} \tag{5.1}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(\widehat{x_i} - x_i)^2}{n}} \tag{5.2}$$

Where:

x_i is the actual rating.

$\hat{x}_i$ is the predicted rating.

n is the total amount of rating data in the testing dataset.

Therefore, lower values in both metrics indicate higher prediction accuracy.

5.2. Implementation Details

The proposed MCoMC recommendation algorithm has been developed and implemented using the Python programming language, in conjunction with the Keras 2.11.0 and TensorFlow frameworks.

In order to find the best parameter configurations, the values of parameter have been established as fixed constants following empirical evaluations. The process involves carefully analyzing the results of the evaluations and selecting the values that would give the best overall performance.

The DNN architecture used for training the MCoMCRS datasets features an input layer with 64 neurons and a three-layer hidden structure (128→128→64). The network utilizes several predefined hyperparameters such as batch size of 50, the activation function ReLU, and a learning rate of 0.01. Moreover, the DNN model incorporates a dropout regularization rate of 0.2 to deal with overfitting in the training process. For both (TripAdvisor and ITM-Rec) the MCoMCRS datasets, the training was conducted over a standardized duration of 150 epochs. A single neuron is used at output layer of network which is responsible for generating overall rating score. The details are shown in **Table *5.2***. To optimize the performance and improve the accuracy of the predictions, the DNN model employs MAE and the RMSE as the loss functions.

Table 5.2. *DNN settings*

Datasets	Parameters	Values
TripAdvisor/ITM-Rec Dataset	Input layer	64
	Hidden layers	(128→128→64)
	Output Layer	1
	Batch Size	50
	Epoch	150

The proposed DNN-based MCoMCRS model is visually described with the help of a flowchart in **Figure *5.1***. It provides a comprehensive overview of the model's operational framework.

The training of this DNN model includes multiple key stages. First, we preprocessed data and partitioned it into suitable training and testing subsets. After that, the model’s inputs are initialized, and the training process starts. On the completion of training, the performance of the model is evaluated. To achieve the minimum error threshold, iterative hyperparameter tuning is performed. This repetitive process ensures the continuous refinement of the model.

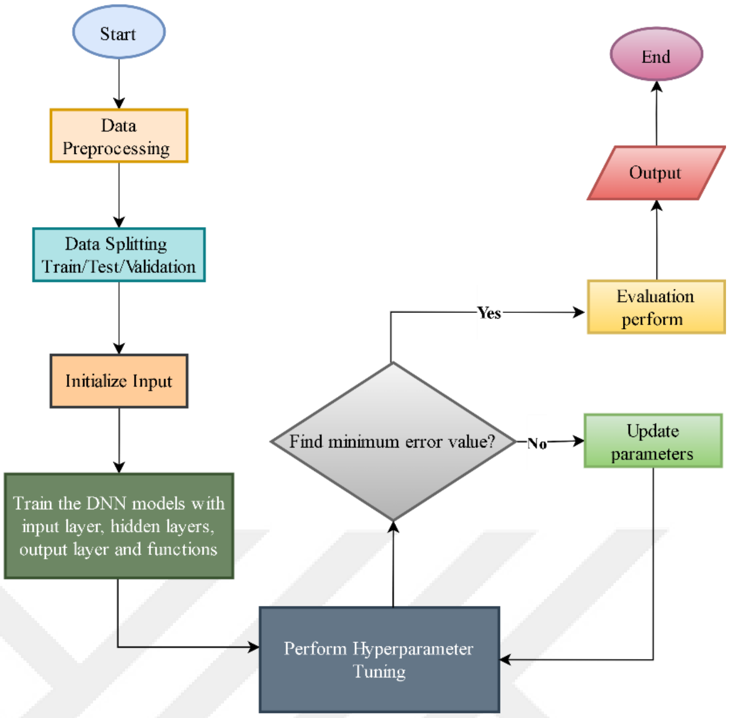

Figure 5.1. *Flowchart of the DNN-based MCoMCRS.*

5.3. Experimental Methodology

To validate the effectiveness of the proposed MCoMCRS model, a set of experiments is performed on tourism and educational datasets. This section provides benchmarking against five distinct baselines within the CARS and MCRS systems.

i. Matrix Factorization is used in the baseline SCRS (Single Criteria Recommender System) as defined in (Koren et al., 2009).

ii. In the MCRS, MF is used to predict individual criterion ratings and to determine the weight of each criterion linear regression is used (Majumder et al., 2017). After that, these criterion-specific ratings are combined using linear regression to predict the overall rating..

iii. Including single-context(Single-Context Recommender System, SCORS) and multi-context (MultiContext Recommender System, MCoRS) both scenarios have been considered for CARS. The models selected for comparison include Item Splitting-BiasdMF, User Splitting-BiasdMF, User Item Splitting-BiasdMF, CAMF (Context-aware Matrix Factorization), CAMF-C, CAMFCI (CAMF Context-Item), and CAMF-CU (CAMF Context-User), all referenced from (Zheng et al., 2015).

iv. To determine if a DNN-based integration of CARS and MCRS outperformed existing DL techniques in terms of predictive accuracy, separate implementations of CARS and MCRS were carried out as specified (Nassar et al., 2020a).

6. RESULTS AND DISCUSSION

The main objective of this study was to develop a novel rating prediction model that integrates CARS and MCRS based on multi-context principles to enhance prediction accuracy. Besides the development of this integrated model, the other primary task was to evaluate whether this integration was capable of outperforming traditional RS, including SCRS, MCRS, and CARS. The comparative analysis, presented in Tables 6.1 and 6.2, evaluated two specific models: (i) a DNN-integrated single-context MCRS and (ii) a DNN-integrated multi-context MCRS. RMSE and MAE were used as performance evaluation metrics. The datasets used in experiments are TripAdvisor and ITM-Rec. It is obvious in **Table *6.1*** and **Table *6.2*** that the proposed models are more effective with their lower values of MAE and RMSE.

Table 6.1. *Comparative performance of proposed models with various predictive models on TripAdvisor*

Algorithms	Models	MAE	RMSE
MF	SCRS	1.4955	1.7603
MF_LR	MCRS	1.1892	1.3698
User-Splitting	SCoRS	0.8339	1.0860
Item-Splitting		0.8443	1.0884
UI-Splitting		0.8416	1.0822
Biased MF		0.8268	1.0599
CAMF-C		0.8791	1.1163
CAMF-CI		0.9048	1.2295
CAMF-CU		0.8720	1.1349
User-Splitting	MCoRS	0.8384	1.0845
Item-Splitting		0.8478	1.0892
UI-Splitting		0.8517	1.1076
Biased MF		0.8565	1.0907
CAMF-C		0.8447	1.0637
CAMF-CI		0.9309	1.2435
CAMF-CU		0.9994	1.3455
DL-CARS	DL	0.7855	1.0806
DL-MCRS		0.5942	0.7403
SCoMCRS		0.4994	0.6759
MCoMCRS	Proposed Model	**0.4737**	**0.6495**

Table 6.2. *Comparative performance of proposed models with various predictive models on ITM-Rec*

Algorithms	Models	MAE	RMSE
MF	SCRS	1.4563	1.7195
MF_LR	MCRS	1.3700	1.5695
User-Splitting	SCoRS	1.1003	1.4367
Item-Splitting		1.0889	1.4213
UI-Splitting		1.0864	1.4268
Biased MF		1.0911	1.4376
CAMF-C		1.1420	1.4644
CAMF-CI		1.0815	1.4130
CAMF-CU		1.1400	1.4823
User-Splitting	MCoRS	1.1101	1.4550
Item-Splitting		1.0519	1.3927
UI-Splitting		1.1074	1.4516
Biased MF		1.1097	1.4484
CAMF-C		1.0705	1.3950
CAMF-CI		1.0598	1.3872
CAMF-CU		1.0789	1.4334
DL-CARS	DL	1.2137	1.5516
DL-MCRS		0.6340	0.8466
SCoMCRS		0.6232	0.8436
MCoMCRS	Proposed Model	**0.6054**	**0.8399**

Both dataset’s results demonstrate that the proposed MCoMCRS model scored the lowest MAE and RMSE. With such consistent results across two different data sources - one from TripAdvisor, and the other from an educational context - we can conclude that the model is both robust and adaptable. This suggests that the underlying methodology of MCoMCRS represents the nuances of user preferences and item characteristics across diverse domains in an effective way, possibly through advanced feature handling or sophisticated interaction modelling.

The performance of deep learning-based model DL-MCRS and Single-Context Multi-criteria Recommender System (SCoMCRS) is also excellent, particularly when compared to traditional matrix factorization methods like MF-LR and MF-MF. Due to deep learning models' ability to capture nonlinear relationships and complex patterns in the data, traditional linear models miss this difference in performance. These models have shown high performance, particularly when used with the TripAdvisor dataset. It indicates that deep learning approaches are more capable at capturing complex, non-obvious patterns in user-item interactions.

Different strategies for splitting items, users, and UIs have resulted in varying levels of model success in SCoRS and MCoRS, suggesting that data partitioning and processing can significantly affect the performance of the model. These strategies have improved upon basic matrix factorization approaches, but they are still unable to match the efficacy of deep learning models. Because they are unable to fully represent the interactions between users and items.

CAMF variants with context-aware mechanisms show mixed results. Depending on the type of data and the specific context, their performance varies between datasets. Therefore, context-aware models would need to be adjusted to the particular features and characteristics of the dataset.

There is less performance gap among models in the ITM-Rec dataset, indicating that datasets are not structured exactly like TripAdvisor data. ITM-Rec datasets might be more homogeneous or less sparse because the results consistently show the same level of accuracy on different models.

It has been observed that most models perform similarly across both datasets, indicating they are robust and generalizable. DL-CARS in the ITM-Rec dataset, the exceptions observed raise important questions about possible overfitting of the model or how it adapts to different types of user-item interaction data.

Therefore, hyper-parameter tuning is performed on both datasets with the proposed MCoMCRS to prevent overfitting and underfitting. The procedure involves tweaking the method's parameters, such as the learning rate, training epochs, number of layers, etc., to see if the performance changes. After that, choose a model that contains the fewest errors.

This study uses a three-layered model, as explained in Section 5. There are 64 neurons in the input layer, (128→128→64) neurons in the three hidden layers, and 1 neuron in the output layer. "ReLU" activation function is used for the input and hidden layers. Our model uses a deep learning framework like Keras to build a sequential neural network. This model has several dense (fully connected) layers and dropout layers. Image 6.1 illustrates the interconnections between layers in the model. The number of parameters is calculated by multiplying the number of inputs by the number of outputs and adding the number of biases in the layer. Based on the calculation, the input layer has ((40 × 64) + 64 = 2560) parameters. To prevent overfitting, the dropout layer with a dropout rate of 0.2 randomly sets 20% of input units to 0 during training. There are three hidden layers with the parameters ((64×128) + 128 = 8320), ((128×128) + 128 = 16512) and ((128 × 64) + 64 = 8256) respectively. The output layer has (64 ×1) + 1 = 65 parameters. The total parameters for this model are 35713. We trained parameters using Keras's "model.fit()" method. The model.summary() method produces the summary output shown in **Figure *6.1***.

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 64)	2560
dropout (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 128)	8320
dropout_1 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 128)	16512
dropout_2 (Dropout)	(None, 128)	0
dense_3 (Dense)	(None, 64)	8256
dense_4 (Dense)	(None, 1)	65

Total params: 35713 (139.50 KB)
Trainable params: 35713 (139.50 KB)
Non-trainable params: 0 (0.00 Byte)

Figure 6.1. *Model summary*

As part of our efforts to evaluate the generalization ability of our model, we conducted a series of experimental studies with multiple learning rates comparing the training loss to validation loss over a number of epochs for both datasets, as illustrated in **Figure *6.2*** and **Figure *6.3***. These images clearly show how the learning rate affects the training process. There is a difference in convergence speeds and stability of loss values resulting from different learning rates. Loss curves are shown with multiple learning rates between 0.001 and 0.5. It is observed that the proposed model on both datasets shows the loss decreases rapidly during the initial epochs and stabilizes thereafter. In particular, the proposed model trained with a learning rate of 0.01 attains the lowest stable loss, signifying an optimal balance between convergence speed and stability. However, the higher learning rate of 0.5 initially reduces the loss rapidly. It causes erratic behavior and a higher final loss, suggesting overshooting. In order to ensure efficient and effective model training, it is imperative to choose an appropriate learning rate.

Also, we can see from these plots that the gap between training and validation loss remains stable as the number of epochs increases, indicating our model has robust generalizability.

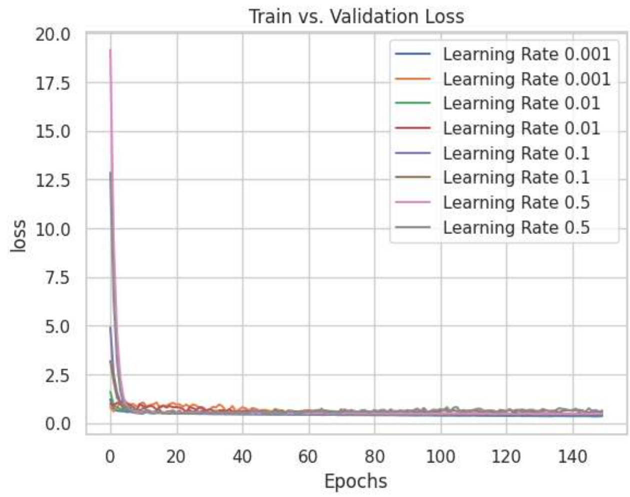

Figure 6.2. *MCoMCRS on TripAdvisor: plot of the training vs validation loss with multiple learning rates*

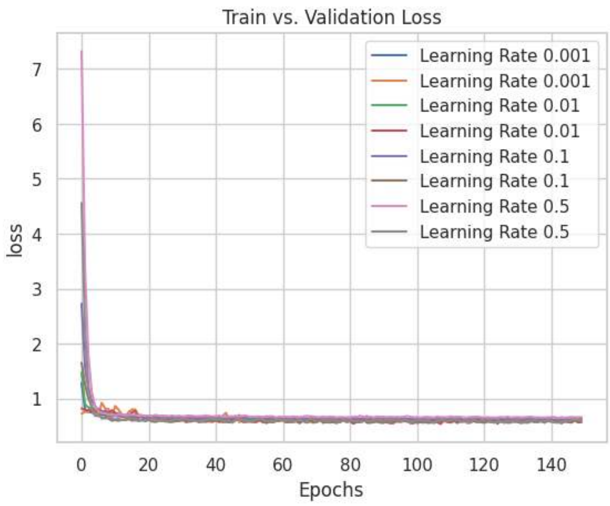

Figure 6.3. *MCoMCRS on ITM-Rec: plot of the training vs validation loss with multiple learning rates*

In **Figure *6.2*** and **Figure *6.3***, the MCoMCRS model is represented, showing the validation and training losses for the TripAdvisor and ITM-Rec datasets with multiple learning rates. The X-axis represents the number of epochs while the Y-axis indicates the loss levels. It is observed that both training and validation losses in our model (with 0.01 learning rate), decreased sharply within the initial 20 epochs and continue to decline more gradually up to 150 epochs. The low gap between the training and validation losses and their concurrent reduction suggests effective handling of overfitting.

We used dropout regularization, batch normalization, and early stopping to reduce the overfitting in our deep neural network. During training, randomly dropping units prevents the model from becoming too dependent on any specific one or few neurons. Batch Normalization (BN) is an optimization technique that normalizes a layer's input to stabilize and speed up the training process of the network. Finally, early stopping monitors the validation loss and stops the training process if the loss increases.

The effectiveness of these techniques is further supported by the consistent decrease in loss across both datasets without significant spikes in validation loss. The graphical analysis confirms our model's high accuracy, which validates the mitigation of overfitting risks. Thus, ensuring the reliability and performance of the model across varied datasets.

In general, the results of the proposed MCoMCRS model indicate that deep learning models can be used efficiently to deal with real-world complex datasets. The MCoMCRS is significantly more accurate in predicting outcomes than all other groups with the lowest MAE and RMSE. Using multi-context with multi-criteria on TripAdvisor dataset, the proposed technique significantly improves recommendation accuracy, surpassing all other algorithms in different categories as demonstrated in **Figure *6.4*** and **Figure *6.5***. Similarly, the proposed MCoMCRS shows the lowest MAE and RMSE in **Figure *6.6*** and **Figure *6.7*** for the ITM-Rec dataset. It is clear that both TripAdvisor and Educational datasets perform consistently better with our proposed algorithm, demonstrating its versatility and adaptability. Due to its flexible architecture and universal applicability of its preprocessing steps, our model can be applied to a wide range of domains. Our experiments were conducted in the travel and educational sectors, but the methods we employed are applicable to a wide range of scenarios that require personalized recommendations, such as healthcare for personalized patient recommendations or e-commerce for product recommendations that are dynamic.

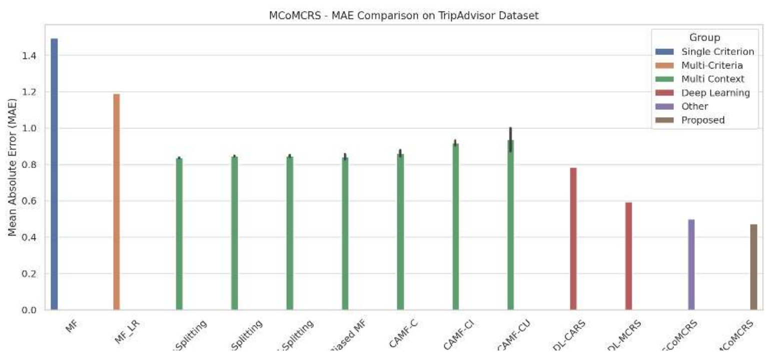

Figure 6.4. *MCoMCRS performance with other algorithms on TripAdvisor dataset, in terms of MAE*

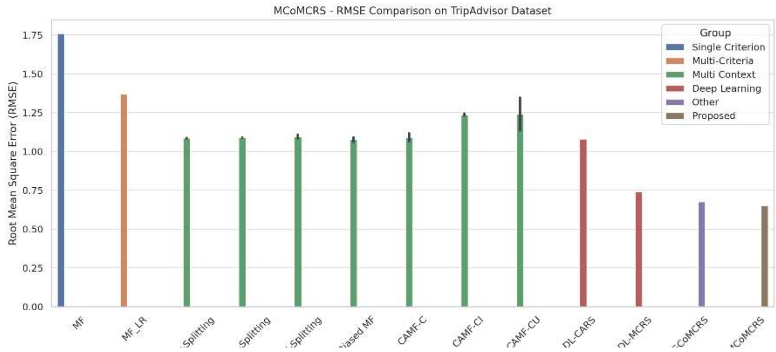

Figure 6.5. *MCoMCRS performance with other algorithms on TripAdvisor dataset, in terms of RMSE*

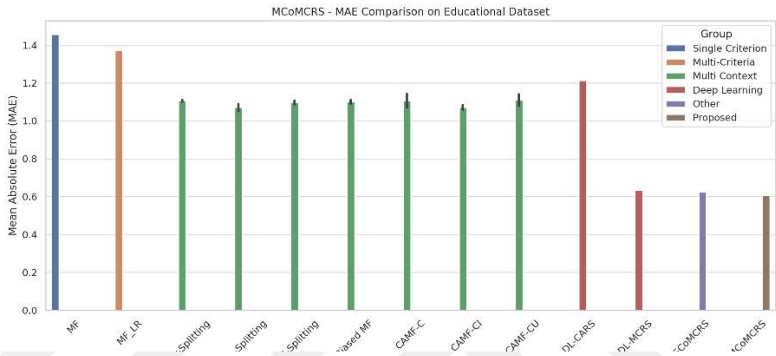

Figure 6.6. *MCoMCRS performance with other algorithms on ITM-Rec dataset, in terms of MAE*

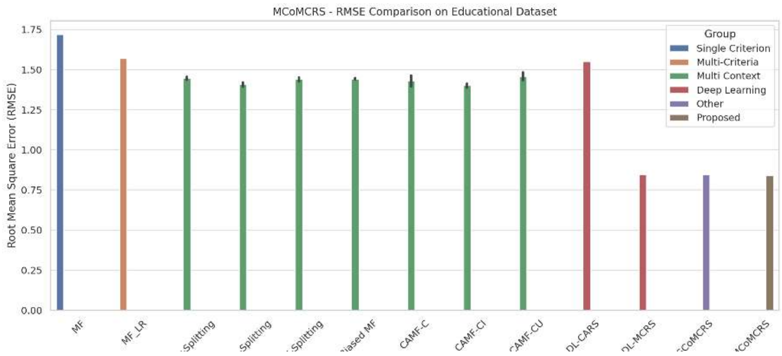

Figure 6.7. *MCoMCRS performance with other algorithms on ITM-Rec dataset, in terms of RMSE*

Using these findings, we highlight not only how dataset-specific characteristics must be considered when developing and selecting models, but also how the model can be extended to other domains, enabling us to test its flexibility and effectiveness in different contexts and criteria. The factors contributing to the superior performance of MCoMCRS would be valuable for future research. By gaining a deeper understanding of the model's inner workings, practical strategies could be developed to handle various scenarios of recommendation.

A more customized and efficient recommendation system could also be created by probing the reasons underlying the inconsistent performance of context-aware models across datasets and the apparent underperformance of certain strategies. Consequently, these findings contribute significantly to recommendation systems, showing the advantages of selecting models based on dataset characteristics and showing the potential of deep learning methods.

7. CONCLUSION AND FUTURE WORK

Deep learning techniques are used in recommender systems as a modern approach to improving recommendations. Extensive research has leveraged deep learning-based approaches to enhance recommendations within individual multi-criteria and context-aware recommender systems. However, the integration of contextual information with multi-criteria ratings within deep learning-based recommender systems is still in its infancy. This study investigates the potential for deep learning techniques to incorporate contextual data into multi-criteria recommender systems that leverage multiple contextual factors, as existing research has predominantly combined single contextual elements with traditional machine learning methods. We specifically used the DNN model to predict the multi-criteria ratings based on multi-context. We have conducted extensive empirical studies and experiments on a diverse range of real-world datasets to comprehensively evaluate the performance and effectiveness of our novel recommendation methods. The results of this thorough analysis demonstrate that our state-of-the-art approach significantly outperforms the existing leading methods in terms of generating high-quality, personalized, and tailored recommendations that cater to the unique preferences and needs of individual users.

This is the first attempt to use DL methods for combining multiple contextual conditions into MCRS. The goal of this integration is to provide a more comprehensive understanding of user preferences which may lead to enhanced recommendation accuracy. Our research work will be compared with other DL-based methods such as autoencoders, CNN and RNN.

In contrast to using separate models for predicting individual multi-criteria ratings, we used a single DNN model for the MCRS. The current approach captures the complex relationships and interdependencies between different criteria. Future work will be extended using multiple models to predict multi-criteria ratings. This may offer deeper insights into the reasons behind user preferences, rather than relying solely on the overall rating.

Furthermore, the DL-based MCoMCRS will be integrated with other well-known recommendation approaches, such as Group-aware Recommender System (GARS) and Personality-aware Recommender Systems (PARS). By integrating these established approaches with the proposed model, researchers will be able to develop a more robust and versatile recommendation framework that can meet the diverse needs of the users.

Recognizing the inherent challenges of data sparsity and the cold start problems, which are common in recommender systems and could impact prediction accuracy, our subsequent efforts will address these issues. Future developments will include the implementation of advanced learning algorithms and novel strategies specifically designed to manage these challenges, ensuring that our system provides superior.

Additionally, the proposed model can also be evaluated and validated by applying it to other real-world datasets with diverse and multifaceted features from various disciplines. In this way, the model's effectiveness and adaptability can be evaluated more comprehensively.

REFERENCES

Abinaya, S., & Devi, M. K. K. (2021). Enhancing Top-N Recommendation Using Stacked Autoencoder in Context-Aware Recommender System. *Neural Processing Letters*, *53*(3), 1865–1888. https://doi.org/10.1007/s11063-021-10475-0

Abowd, G. D., Dey, A. K., Brown, P. J., Davies, N., Smith, M., & Steggles, P. (1999). *Towards a Better Understanding of Context and Context-Awareness BT - Handheld and Ubiquitous Computing* (H.-W. Gellersen (ed.); pp. 304–307). Springer Berlin Heidelberg.

Adomavicius, G., & Kwon, Y. (2007). New recommendation techniques for multicriteria rating systems. *IEEE Intelligent Systems*, *22*(3), 48–55. https://doi.org/10.1109/MIS.2007.58

Adomavicius, G., Manouselis, N., & Kwon, Y. (2011). Multi-criteria recommender systems. In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor (Eds.), *Recommender systems handbook* (pp. 769–803). Springer US. https://doi.org/10.1007/978-0-387-85820-3_24

Adomavicius, G., & Tuzhilin, A. (2005a). Incorporating Contextual Information in Recommender Systems Using a Multidimensional Approach University of Minnesota University of Connecticut. *ACM Transactions on Information Systems*, *23*(1), 103–145.

Adomavicius, G., & Tuzhilin, A. (2005b). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, *17*(6), 734–749. https://doi.org/10.1109/TKDE.2005.99

Afzal, I., Yilmazel, B., & Kaleli, C. (2024). An Approach for Multi-Context-Aware Multi-Criteria Recommender Systems Based on Deep Learning. *IEEE Access*, *12*(July), 99936–99948. https://doi.org/10.1109/ACCESS.2024.3428630

Ashley-Dejo, E., Ngwira, S. M., & Zuva, T. (2016). A context-aware proactive recommender system for tourist. *2016 International Conference on Advances in Computing and Communication Engineering (ICACCE)*, 271–275.

Batmaz, Z., & Kaleli, C. (2019). AE-MCCF: an autoencoder-based multi-criteria recommendation algorithm. *Arabian Journal for Science and Engineering*, *44*, 9235–9247. https://doi.org/10.1007/s13369-019-03946-z

BATMAZ, Z., & KALELİ, C. (2022). A new similarity-based multicriteria recommendation algorithm based onautoencoders. *Turkish Journal of Electrical Engineering and Computer Sciences*, *30*(3), 855–870. https://doi.org/10.55730/1300-0632.3815

Beutel, A., Covington, P., Jain, S., Xu, C., Li, J., Gatto, V., & Chi, E. H. (2018). Latent cross: Making use of context in recurrent recommender systems. *WSDM 2018 - Proceedings of the 11th ACM International Conference on Web Search and Data Mining*, *2018-Febua*, 46–54. https://doi.org/10.1145/3159652.3159727

Bilge, A., & Kaleli, C. (2014). A multi-criteria item-based collaborative filtering framework. *2014 11th International Joint Conference on Computer Science and*

Software Engineering (JCSSE), 18–22. https://doi.org/10.1109/JCSSE.2014.6841835

Bobadilla, J., Ortega, F., Hernando, A., & Gutiérrez, A. (2013). Knowledge-Based Systems Recommender systems survey. *Knowledge-Based Systems*, *46*, 109–132. https://doi.org/10.1016/j.knosys.2013.03.012

Boppana, V., & Sandhya, P. (2021). Web crawling based context aware recommender system using optimized deep recurrent neural network. *Journal of Big Data*, *8*(1). https://doi.org/10.1186/s40537-021-00534-7

Dridi, R., Tamine, L., & Slimani, Y. (2019). Context-aware multi-criteria recommendation based on spectral graph partitioning. *30th International Conference on Database and Expert Systems Applications*, 211–221.

Dridi, R., Tamine, L., & Slimani, Y. (2022). Exploiting context-awareness and multi-criteria decision making to improve items recommendation using a tripartite graph-based model. *Information Processing & Management*, *59*(2), 102861. https://doi.org/10.1016/j.ipm.2021.102861

Esteban, A., Zafra, A., & Romero, C. (2020). Helping university students to choose elective courses by using a hybrid multi-criteria recommendation system with genetic optimization. *Knowledge-Based Systems*, *194*, 105385. https://doi.org/10.1016/j.knosys.2019.105385

Gasparic, M., & Ricci, F. (2015). Modeling context-aware command recommendation and acceptance in an IDE. *2015 IEEE/ACM 2nd International Workshop on Context for Software Development*, 1–5.

Goldberg, D., Nichols, D., Oki, B. M., & Terry, D. (1992). Using collaborative filtering to Weave an Information tapestry. *Communications of the ACM*, *35*(12), 61–70. https://doi.org/10.1145/138859.138867

Gunes, I., Kaleli, C., Bilge, A., & Polat, H. (2014). Shilling attacks against recommender systems: a comprehensive survey. *Artificial Intelligence Review*, *42*(4), 767–799. https://doi.org/10.1007/S10462-012-9364-9

Gupta, S., & Kant, V. (2020). An aggregation approach to multi-criteria recommender system using genetic programming. *Evolving Systems*, *11*(1), 29–44. https://doi.org/10.1007/s12530-019-09296-3

Hassan, M., & Hamada, M. (2017). A neural networks approach for improving the accuracy of multi-criteria recommender systems. *Applied Sciences (Switzerland)*, *7*(9). https://doi.org/10.3390/app7090868

Herlocker, J. L., Konstan, J. A., & Riedl, J. (2000). Explaining collaborative filtering recommendations. *Proceedings of the ACM Conference on Computer Supported Cooperative Work*, 241–250. https://doi.org/10.1145/358916.358995

Hien, N. L. H., Van Huy, L., Huu, H., & Nguyen Van Hieu, M. (2024). A Deep Learning Model for Context Understanding in Recommendation Systems. *Informatica (Slovenia)*, *48*(1), 31–44. https://doi.org/10.31449/inf.v48i1.4475

Jawarneh, I. M. Al, Bellavista, P., Corradi, A., Foschini, L., Montanari, R., Berrocal, J., & Murillo, J. M. (2020). A Pre-Filtering Approach for Incorporating Contextual Information into Deep Learning Based Recommender Systems. *IEEE Access*, *8*, 40485–40498. https://doi.org/10.1109/ACCESS.2020.2975167

Jeong, S. Y., & Kim, Y. K. (2022). Deep Learning-Based Context-Aware Recommender System Considering Contextual Features. *Applied Sciences (Switzerland)*, *12*(1). https://doi.org/10.3390/app12010045

Jeong, S. Y., & Kim, Y. K. (2023). Deep Learning-Based Context-Aware Recommender System Considering Change in Preference. *Electronics (Switzerland)*, *12*(10). https://doi.org/10.3390/electronics12102337

Jhalani, T., Kant, V., & Dwivedi, P. (2016). A linear regression approach to multi-criteria recommender system. *Data Mining and Big Data: First International Conference, DMBD 2016, Bali, Indonesia, June 25-30, 2016. Proceedings 1*, 235–243.

Kant, V., Jhalani, T., & Dwivedi, P. (2018). Enhanced multi-criteria recommender system based on fuzzy Bayesian approach. *Multimedia Tools and Applications*, *77*(10), 12935–12953. https://doi.org/10.1007/s11042-017-4924-2

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, *42*(8), 30–37. https://doi.org/10.1109/MC.2009.263

Liao, J., Liu, T., Liu, M., Wang, J., Wang, Y., & Sun, H. (2018). Multi-Context Integrated Deep Neural Network Model for Next Location Prediction. *IEEE Access*, *6*(c), 21980–21990. https://doi.org/10.1109/ACCESS.2018.2827422

Lops, P., de Gemmis, M., & Semeraro, G. (2011). Content-based Recommender Systems: State of the Art and Trends. *Recommender Systems Handbook*, 73–105. https://doi.org/10.1007/978-0-387-85820-3_3

Majumder, G. S., Dwivedi, P., & Kant, V. (2017). Matrix factorization and regression-based approach for multi-criteria recommender system. *Information and Communication Technology for Intelligent Systems (ICTIS 2017)*, *1*(2), 103–110.

Manouselis, N., & Costopoulou, C. (2007). Analysis and classification of multi-criteria recommender systems. *World Wide Web*, *10*(4), 415–441. https://doi.org/10.1007/s11280-007-0019-8

Manouselis, N., Verbert, K., Drachsler, H., & Santos, O. C. (2010). Workshop on recommender systems for Technology Enhanced Learning. *RecSys'10 - Proceedings of the 4th ACM Conference on Recommender Systems*, *17*(6), 377. https://doi.org/10.1145/1864708.1864797

Monti, D., Rizzo, G., & Morisio, M. (2021). A systematic literature review of multicriteria recommender systems. *Artificial Intelligence Review*, *54*, 427–468. https://doi.org/10.1007/s10462-020-09851-4

Nassar, N., Jafar, A., & Rahhal, Y. (2020a). A novel deep multi-criteria collaborative filtering model for recommendation system. *Knowledge-Based Systems*, *187*. https://doi.org/10.1016/j.knosys.2019.06.019

Nassar, N., Jafar, A., & Rahhal, Y. (2020b). Multi-criteria collaborative filtering recommender by fusing deep neural network and matrix factorization. *Journal of Big Data*, *7*(1). https://doi.org/10.1186/s40537-020-00309-6

Nguyen, T.-N., & Nguyen, A.-T. (2015). Weighing the role of multi-criteria communities for recommender systems. *International Journal of Intelligent Engineering Informatics*, *3*(4), 330–348.

Nilashi, M., Ahani, A., Esfahani, M. D., Yadegaridehkordi, E., Samad, S., Ibrahim, O., Sharef, N. M., & Akbari, E. (2019). Preference learning for eco-friendly hotels recommendation: A multi-criteria collaborative filtering approach. *Journal of Cleaner Production*, *215*, 767–783. https://doi.org/10.1016/j.jclepro.2019.01.012

Odić, A., Tkalčič, M., Tasič, J., & Košir, A. (2012). Relevant Context in a Movie Recommender System: Users' Opinion vs. Statistical Detection. *CARS-2012. Context-Aware Recommender Systems 2012: Proceedings of the 4th Workshop on Context-Aware Recommender Systems in Conjunction with the 6th ACM Conference on Recommender Systems (RecSys 2012)*, *889*.

Philosophy, D. O. F. (2022). *CONTEXT AWARE RECOMMENDER SYSTEMS USING DEEP NEURAL NETWORK Thesis submitted in fulfillment for the requirements for the Degree of*. *June*.

Ramchandani, L., Patel, S., & Suthar, K. (2020). A Multiple Criteria-Based Context-Aware Recommendation System for Agro-Cloud. In H. Sharma, K. Govindan, R. C. Poonia, S. Kumar, & W. M. El-Medany (Eds.), *Advances in Computing and Intelligent Systems* (pp. 557–564). Springer Singapore. https://doi.org/10.1007/978-981-15-0222-4_53

Ranjbar Kermany, N., & Alizadeh, S. H. (2017). A hybrid multi-criteria recommender system using ontology and neuro-fuzzy techniques. *Electronic Commerce Research and Applications*, *21*, 50–64. https://doi.org/https://doi.org/10.1016/j.elerap.2016.12.005

Rodríguez, S. V., & Viktor, H. L. (2013). A personalized location aware multi-criteria recommender system based on context-aware user preference models. *IFIP Advances in Information and Communication Technology*, *412*, 30–39. https://doi.org/10.1007/978-3-642-41142-7_4

Safoury, L., & Salah, A. (2013). Exploiting User Demographic Attributes for Solving Cold-Start Problem in Recommender System. *Lecture Notes on Software Engineering*, *1*(3), 303–307. https://doi.org/10.7763/lnse.2013.v1.66

Santana, I. A. P., & Domingues, M. A. (2020). A systematic review on context-aware recommender systems using deep learning and embeddings. *ArXiv Preprint ArXiv:2007.04782*.

Sassi, I. Ben, Yahia, S. Ben, & Liiv, I. (2021). MORec: At the crossroads of context-aware and multi-criteria decision making for online music recommendation. *Expert Systems with Applications*, *183*, 115375. https://doi.org/10.1016/j.eswa.2021.115375

Sattar, A., & Bacciu, D. (2023). Graph Neural Network for Context-Aware Recommendation. *Neural Processing Letters*, *55*(5), 5357–5376. https://doi.org/10.1007/s11063-022-10917-3

Schafer, J. Ben, Frankowski, D., Herlocker, J., & Sen, S. (2007). Collaborative filtering recommender systems. In P. Brusilovsky, A. Kobsa, & W. Nejdl (Eds.), *The adaptive web: methods and strategies of web personalization* (pp. 291–324). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-72079-9_9

Shambour, Q. (2021). A deep learning based algorithm for multi-criteria recommender systems. *Knowledge-Based Systems*, *211*, 106545.

https://doi.org/10.1016/j.knosys.2020.106545

Shambour, Q., Hourani, M., & Fraihat, S. (2016). An Item-based Multi-Criteria Collaborative Filtering Algorithm for Personalized Recommender Systems. *International Journal of Advanced Computer Science and Applications*, *7*(8). https://doi.org/10.14569/ijacsa.2016.070837

Shani, G., & Gunawardana, A. (2011). *Evaluating Recommendation Systems BT - Recommender Systems Handbook* (F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor (eds.); pp. 257–297). Springer US. https://doi.org/10.1007/978-0-387-85820-3_8

Tallapally, D., Sreepada, R. S., Patra, B. K., & Babu, K. S. (2018). User preference learning in multi-criteria recommendations using stacked auto encoders. *RecSys 2018 - 12th ACM Conference on Recommender Systems*, 475–479. https://doi.org/10.1145/3240323.3240412

Toledo, R. Y., Alzahrani, A. A., & Martinez, L. (2019). A food recommender system considering nutritional information and user preferences. *IEEE Access*, *7*, 96695–96711.

Unger, M., Tuzhilin, A., & Livne, A. (2020). Context-Aware Recommendations Based on Deep Learning Frameworks. *ACM Transactions on Management Information Systems*, *11*(2). https://doi.org/10.1145/3386243

Vu, S.-L., & Le, Q.-H. (2023). A Deep Learning Based Approach for Context-Aware Multi-Criteria Recommender Systems. *Computer Systems Science and Engineering*, *44*(1), 471–483.

Wasid, M., & Ali, R. (2018). An improved recommender system based on multi-criteria clustering approach. *Procedia Computer Science*, *131*, 93–101. https://doi.org/10.1016/j.procs.2018.04.190

Wu, T., Sun, F., Dong, J., Wang, Z., & Li, Y. (2022). Context-aware session recommendation based on recurrent neural networks. *Computers and Electrical Engineering*, *100*, 107916. https://doi.org/https://doi.org/10.1016/j.compeleceng.2022.107916

Yeung, K. F., Yang, Y., & Ndzi, D. (2012). A proactive personalised mobile recommendation system using analytic hierarchy process and Bayesian network. *Journal of Internet Services and Applications*, *3*, 195–214.

Zheng, Y. (2017). Criteria chains: a novel multi-criteria recommendation approach. *Proceedings of the 22nd International Conference on Intelligent User Interfaces*, 29–33. https://doi.org/10.1145/3025171.3025215

Zheng, Y. (2019a). Multi-criteria Recommendations by Using Criteria Preferences as Contexts. In T. A. Majchrzak, C. Mateos, F. Poggi, & T.-M. Grønli (Eds.), *Towards Integrated Web, Mobile, and IoT Technology* (Vol. 347, pp. 21–35). Springer.

Zheng, Y. (2019b). Utility-based multi-criteria recommender systems. *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, 2529–2531. https://doi.org/10.1145/3297280.3297641

Zheng, Y. (2022). DeepCARSKit: A deep learning based context-aware recommendation library[Formula presented]. *Software Impacts*, *13*(April), 100292. https://doi.org/10.1016/j.simpa.2022.100292

Zheng, Y. (2023). *ITM-Rec: An Open Data Set for Educational Recommender Systems*. 7–10. http://arxiv.org/abs/2303.10230

Zheng, Y., & Mobasher, B. (2018). *Context-Aware recommendations*. 1–33.

Zheng, Y., Mobasher, B., & Burke, R. (2015). Carskit: A java-based context-aware recommendation engine. *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, 1668–1671. https://doi.org/10.1109/ICDMW.2015.222

Zheng, Y., Shekhar, S., Jose, A. A., & Rai, S. K. (2019). Integrating context-awareness and multi-criteria decision making in educational learning. *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, 2453–2460. https://doi.org/10.1145/3297280.3297522

APPENDIX

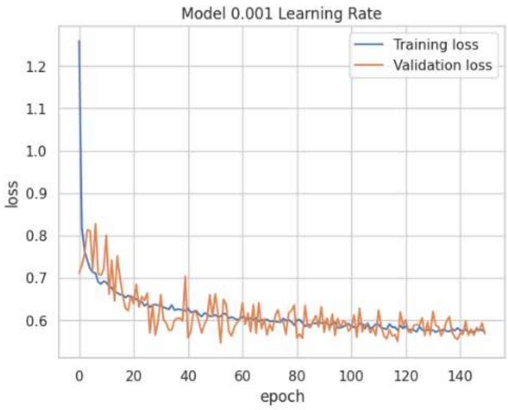

Figure 0.1. *MCoMCRS on ITM: plot of the training vs validation loss with 0.001 learning rates*

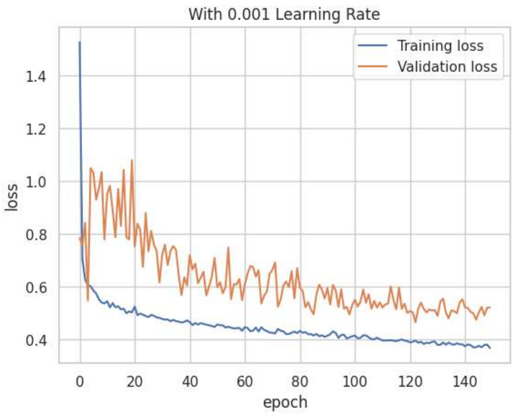

Figure 0.2. *MCoMCRS on TripAdvisor: plot of the training vs validation loss with 0.001 learning rates*

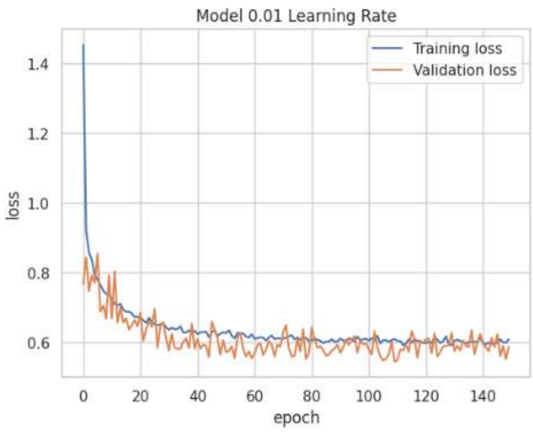

Figure 0.3. *MCoMCRS on ITM: plot of the training vs validation loss with 0.01 learning rates*

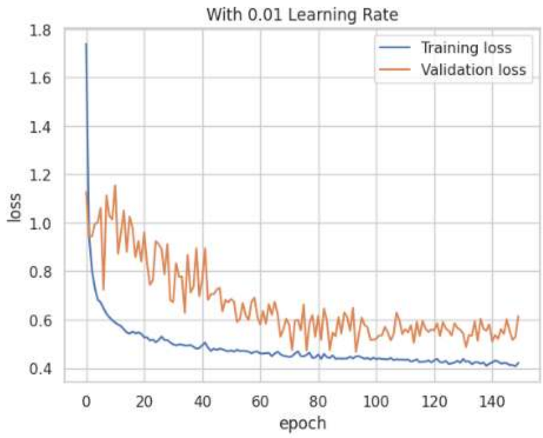

Figure 0.4. *MCoMCRS on TripAdvisor: plot of the training vs validation loss with 0.01 learning rates*

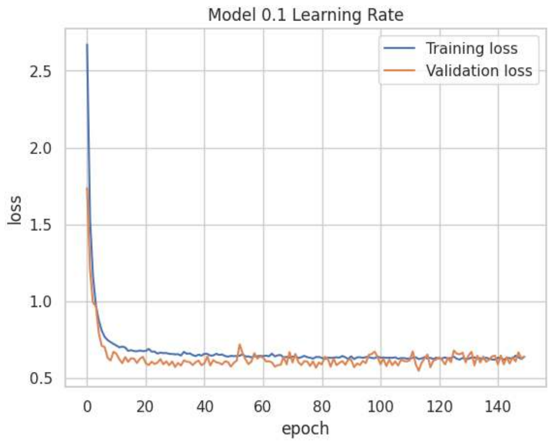

Figure 0.5. *MCoMCRS on ITM: plot of the training vs validation loss with 0.1 learning rates*

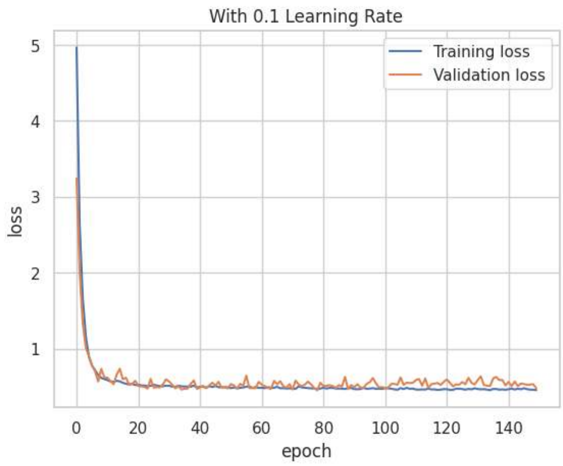

Figure 0.6. *MCoMCRS on TripAdvisor: plot of the training vs validation loss with 0.1 learning rates*

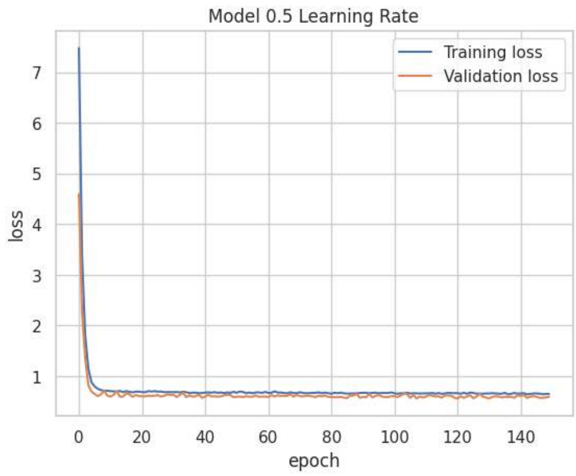

Figure 0.7. *MCoMCRS on ITM: plot of the training vs validation loss with 0.5 learning rates*

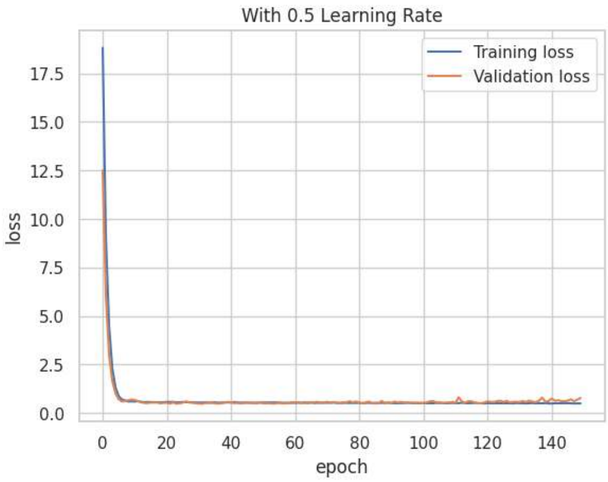

Figure 0.8. *MCoMCRS on TripAdvisor: plot of the training vs validation loss with 0.5 learning rates*

CURRICULUM VITAE

ORCID ID: 0000-0001-8119-1248

Name Surname **: Ifra AFZAL**

Foreign Language **: English**

Education Background:

- Doctorate (2016- 2024), Eskisehir Technical University, Computer Engineering
- MCS (2012 - 2014), COMSATS Institute of Information Technology, Computer Science
- Bachelors (2010 – 2012), Islamia University Bahawalpur, Computer Science

Professional Background:

- Lecturer (2014-2015), Govt College University Faisalabad, Sahiwal
- SESE Teacher (2015-2016), Govt School, Fort Abbas

Publications during PhD. Study:

- Afzal, I., Yilmazel, B., & Kaleli, C. (2024). An Approach for Multi-Context-Aware Multi-Criteria Recommender Systems Based on Deep Learning. *IEEE Access*, *12*(July), 99936–99948. https://doi.org/10.1109/ACCESS.2024.3428630

Awards:

- 2014, Master in Computer Science, Gold Medal, Pakistan