Artificial Intelligence-Driven Recommender Solutions for E-Commerce: A Multidisciplinary Approach for Enhancing Collaborative Filtering Quality

Osama M. M. Alshareet

A Thesis
in the
Concordia Institute
for
Information Systems Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy (Information Systems Engineering)

at Concordia University
Montréal, Québec, Canada

September 2024

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Osama M. M. Alshareet**

Entitled: **Artificial Intelligence-Driven Recommender Solutions for E-Commerce: A Multidisciplinary Approach for Enhancing Collaborative Filtering Quality.**

and submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy (Information and Systems Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

_____ Chair
Dr. Mohsen Ghafouri

_____ External Examiner
Dr. Sudhanshu Joshi

_____ External to Program
Dr. Ciprian Alexandru

_____ Examiner
Dr. Jun Yan

_____ Examiner
Dr. Chun Wang

_____ Supervisor
Dr. Anjali Awasthi

Approved by _____
Dr. Farnoosh Naderkhani
Chair of Department or Graduate Program Director

November 12, 2024

Date of Defence

_____
Dr. Mourad Debbabi, Dean
Gina Cody School of Engineering and Computer Science

Abstract

Artificial Intelligence-Driven Recommender Solutions for E-Commerce: A Multidisciplinary Approach for Enhancing Collaborative Filtering Quality.

Osama M. M. Alshareet, Ph.D.
Concordia University, 2024

AI-driven recommender systems are transforming E-commerce by taking on tasks traditionally handled by human staff, such as sales associates, inventory clerks, and others. They apply machine learning methods to replace human decisions, enhance accuracy and scalability, and improve customer experiences. In this context, significant achievements have been witnessed in recent years in improving collaborative filtering-based recommender systems (CFRSs) through optimizing recall and normalized discounted cumulative gain (NDCG) metrics. Nonetheless, major issues remain that significantly limit the performance and generalization of these systems, such as diversity and novelty in recommendations, fairness, inclusion of long-tail items, the cold start problem, reproducibility, and evaluation overfitting. This study advocates the need for new approaches for addressing these problems comprehensively, moving beyond the traditional optimization metrics (recall and NDCG).

This work is novel in its multidisciplinary approach, integrating principles from systems engineering, software engineering, and TRIZ into the development and optimization of CFRSs. Since systems engineering takes a holistic standpoint, it allows for the reasoning and optimization of user-item interactions. Meanwhile, Software engineering provides several systematic ways and techniques to analyze and improve the functional parts of CFRSs. The TRIZ methodology facilitates the development of innovative solutions and AI tools to eliminate technical contradictions and enhance the performance of CFRSs.

To guide the optimization of CFRSs, the research also uses the ISO/IEC 25010:2011 standards to evaluate the CFRSs thoroughly. These standards evaluate the reliability, usability, performance efficiency, and privacy of the CFRSs against high-quality benchmarks. The evaluation of the results based on real-world data pertaining to E-commerce datasets demonstrates that the recommendation accuracy, diversity, and coverage were improved.

All in all, the current research improves CFRS technology by providing robust, innovative, and user-centric solutions. The proposed multidisciplinary approaches serve as a template for future research and development work. The findings achieved, peer-reviewed, and published in various publications have contributed to the discourse in academics along with practical implementation by creating high-quality CFRSs.

Acknowledgments

It has been a long and fulfilling journey to work on this Ph.D. thesis, and I could not have done it without the encouragement and motivation extended by a huge number of people. Most importantly, I would like to express my appreciation to my supervisor, Professor Anjali Awasthi, for her direction, patience, and unwavering support during my doctoral study. Furthermore, I would like to extend my sincere thanks to Professor Abdessamad Ben Hamza.

I also wish to thank my dissertation committee members and the GPD: Dr. Sudhanshu Joshi, Dr. Ciprian Alecsandru, Dr. Jun Yan, Dr. Mohsen Ghafouri, Dr. Chun Wang, and Dr. Farnoosh Naderkhani for their time, expert comments, and insightful feedback, without which the additional value of my work would not have been possible.

I would like to express my sincere gratitude to the CIISE administrative staff for their invaluable support throughout this work. Special thanks go to Silvie Pasquarelli, Mireille Wahba, Kim Adams, and Laura Oproiu, whose support was immensely helpful. Their dedication, efficiency, and commitment have been instrumental in facilitating the progress of this work.

Thank you all.....

Osama M. M. Alshareet

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter introduces the main concepts and methods related to Recommender Systems(RSs), focusing on CFRSs. It starts by explaining why it is important to consider different dimensions to solve challenges in these systems. Key aspects like the ratings matrix and important priorities for designing such systems are covered. The chapter also examines current solutions, how success is measured, and the main goals of CFRSs.

The chapter also outlines the research problem, pointing out existing issues and gaps. It describes the methods used in the research and highlights its objectives. The awareness levels for handling the challenges in CFRSs are introduced as well. A list of related publications and contributions from the research is provided to show its relevance. Finally, the chapter presents an outline of the thesis to guide readers through its structure.

1.1 Understanding Recommender Systems: A Comprehensive Approach

RSs are tools that influence how users engage with digital platforms. They analyze patterns in user behavior to suggest items a user might like. These systems are used widely in areas like E-commerce, streaming, and social media to give personal suggestions. For example, an online store may recommend products that others with similar tastes have bought. Likewise, streaming platforms can suggest movies or shows, and social networks can recommend new connections. By using CFRS's algorithms, these systems focus on delivering content that matches individual preferences.

In the modern digital age, RSs simplify the process of finding relevant information from the overwhelming choices available. They enhance user experience and make platforms more appealing and engaging. These systems collect data from different sources, such as past user interactions or social links, and sometimes include factors like time or location. Algorithms process this data to find patterns and predict user preferences. This personalized approach makes platforms more user-friendly and competitive.

Still, building effective RSs comes with its challenges. It can be hard to model users' changing preferences while keeping their data safe and private. There is also a risk of creating echo chambers or increasing biases, which can limit the diversity of recommendations. To address this, systems aim to provide a variety of recommendations that are both useful and balanced. More details about the types of RSs are expanded in Section 2.1.1

As RSs evolve, the focus is shifting from just providing relevant suggestions to offering more diverse and unexpected ones. While it's important to recommend items based on users' known interests, it's equally valuable to suggest new things they wouldn't find on their own. This balance between meeting expectations and introducing surprises improves user satisfaction and makes platforms more interesting to use. Addressing this, a comprehensive approach in CFRSs focuses on multiple dimensions:

- **Diversity and Novelty:** Recommendations should be broad enough and include novel items that trigger

users' discovery beyond the common or popular items. [1–3]

- **Fairness:** Partition users and ensure equal representation of diverse preferences, such that no specific interest, nationality, or any other specific demographic data is predominant. [2, 4]
- **Long-Tail Recommendations:** Include some less common items in order to mitigate the risks associated with data sparsity. This mitigates a potential limitation in CFRSs capacity to recommend only popular tastes. [2]
- **Cold Start:** Effective strategies that should support users or items with very limited historical data so they can also have recommendations that are precise and relevant to them. [2, 3]
- **Sparsity:** Sparsity is a problem that requires a more general solution such that it can account for all user tastes and their preferences, notwithstanding the few interactions inside most datasets, which are large in number. [2, 3, 5]
- **Reproducibility:** Ensure that the reliability of the developed CFRSs' solutions is guaranteed in various experimental settings and environments. [6, 7]

This thesis, therefore, elaborates and discusses the development and improvement of CFRSs, which are in need of further enhancements. In particular, the framework has new techniques that are suggested for enhancement, thus raising the state of the art in measurement and criteria as high as possible, with the aim of making a significant impact in the research area. Indeed, this method is expected to increase the quality of the recommendations and to make them more effective; in the large scheme of things, it will lay the foundation for future personalization experience for diverse base of users.

1.2 Impact of AI-Driven Recommender Systems on E-commerce

AI-driven RSs have greatly transformed E-commerce websites, improving customer satisfaction and business performance [8]. The ability to make online shopping experiences more personal through retrieval of data from past purchase information to users' browsing history enables them to recommend products that fit a user's interest. These are some elements that contribute to the multifaceted impacts that AI-driven RSs have in relation to E-commerce:

- **Personalized Shopping Journeys**: Using Machine Learning algorithms, RSs make customer journeys more personalized and boost the user experience by making suggestions for products that are in tune with individual preferences, from workout gear to nutritional supplements [9].
- **Increased Sales and Revenue**: E-commerce sites experience vast sales through recommendation algorithms, which can effectively nudge a user to probable purchases, hence attaining increased sales for both planned and impulse buying [10].
- **Heightened User Engagement and Loyalty**: The ability of machine learning algorithms to predict the most popular products enables E-commerce websites to show trendy products. Accordingly, increases user engagement and customer loyalty, and encourages users to visit more and make more purchases [11].
- **Streamlined Inventory Management**: Using algorithms to predict supply and demand, RSs allow an E-commerce website to optimize stock levels based on user preference and buying behavior analysis, ensuring products that sell are always available [12]. Accordingly, this will significantly facilitate supply chain and inventory management.

- **Targeted Marketing Strategies**: A site like Etsy uses information derived from RSs to run targeted campaigns. Consequently, ensure that users are shown interesting ads and increase the effectiveness of the campaign [13].
- **Cross-Selling and Up-Selling Efficiency**: E-commerce websites exploit such RSs to know what to cross-sell and up-sell; by recommending some complementary or important product, it increases the overall value of the cart [14].
- **Trust and Credibility Through Quality Recommendations**: It is one of the most important rationales for stores like Sephora to award the highest ratings to their products, hence testifying to the fact that the quality of their sellable items can be vouched for, which will, in turn, increase convenience and credibility toward them [15].
- **Discovery of New Products and Categories**: RSs make it possible for users to discover new products and categories available on, for instance, AliExpress. It is through such a manner that shopping experiences are personalized around items that the users might not have found during their search [16].
- **Dynamic Promotion Optimization**: Machine learning algorithms analyze data on user interaction over E-commerce sites and use it for the customization of promotions whereby discounts or other incentives are offered alternately on the items of interest that the users have not yet bought, with the aim of getting more of them to make the purchase and drive the sales higher [17].

Therefore, all the above-mentioned applications of recommender solutions give personal treatment to shopping, and E-commerce sites are powered with personalization in order to maximize strategic marketing, sales, and customer interaction.

1.3 Introduction to Collaborative Filtering

Collaborative filtering (CF) is considered one of the simplest basic concepts in the domain of recommendation systems, offering better predictive accuracy by using common preferences among users. In simple terms, the underlying idea is that, based on history of similar users, one would assume that one of these users would have similar preferences in the future. Therefore, by representing it in this way, it offers a systematic mechanism of how the relationships between users and items can be described and analyzed, so that the complicated patterns of user-item interactions may be understood better. This will not only increase the accuracy of recommendations but also improve the understanding of users' preferences in such a way that more relevant and personalized recommendations can be made.

Let there be a bipartite graph $\mathcal{G}(\mathcal{U}, \mathcal{I}, \mathcal{E})$ that represents users $\mathcal{U} = u_1, u_2, \ldots, u_m$ and items $\mathcal{I} = i_1, i_2, \ldots, i_n$, where $\mathcal{E}$ is the set of edges representing interactions like ratings, purchases, and viewings. This graph representation demonstrates the relationships between users and items, hence providing a structured way to analyze complex data that enhances the accuracy of recommendations.

The general idea with which collaborative filtering can be implemented using such a bipartite representation is through predicting unknown interactions or ratings, r_{ui}, which a user, $u \in \mathcal{U}$, may assign to an item, $i \in \mathcal{I}$. To this end, a fair number of functions have been proposed for purposes of learning $f : \mathcal{U} \times \mathcal{I} \rightarrow \mathbb{R}$, to properly estimate these interactions, thereby affecting in a direct way the effectiveness of the recommendation systems.

1.3.1 Ratings Matrix

Interactions in the framework of collaborative filtering are modeled as a binary ratings matrix $\boldsymbol{R}$, with dimensions $m \times n$, where each entry $\boldsymbol{R}_{u,i} = 1$ if user u has interacted with item i, and $\boldsymbol{R}_{u,i} = 0$ otherwise. This simplifies the interaction data into a matrix of all user actions as positive indications of preference,

regardless of the nature of the interaction. This type of interaction matrix is the heart of a CFRS and acts as the main structure from which user preferences and similarities arise. This clearly differentiates between the items that a user has interacted with and those they have not, so that more personalized recommendations can be made. Graph theory is described in more detail in Section 2.1.4.

1.3.2 Core Priorities in Designing CFRSs

Improving accuracy and diversity of recommendations is considered as core priority in designing CFRSs:

- **Accuracy:** Represents the prediction of how close the interaction of the user-item $\boldsymbol{R}_{ui}$ can be given by the function $f(u, i)$, which calculates the likelihood of interaction between the user u and item i.
- **Diversity:** Recommendations for user u should present a mix of items that are both familiar and new. This approach highlights u's broader interests and promotes exploration.

1.3.3 Current Solution Approaches

There are numerous predictive methodologies to estimate the unknown interactions or ratings of users with items, denoted as $\hat{\boldsymbol{R}}_{ui}$. Similarly to $\boldsymbol{R}$, the matrix $\hat{\boldsymbol{R}}$ has dimensions $m \times n$, where m corresponds to the total number of users and n corresponds to the total number of items. $\hat{\boldsymbol{R}}$ consists of the predicted ratings where each entry $\hat{\boldsymbol{R}}_{ui}$ denotes the estimated interaction strength or preference score that user u would assign to item i, based on the modeling techniques used. Such values need to be predicted in the most accurate manner so that they converge on the actual user preferences, which are often either not known or not observed for non-interactive elements. This is achieved in the following ways:

- **Matrix Factorization**: A decomposition method for a binary rating matrix $\boldsymbol{R}$ into latent factors representing underlying characteristics of users and items. The product of these factors realizes the predicted ratings $\hat{\boldsymbol{R}}_{ui}$ leading to insights into potential user preferences for items.
- **Neighborhood Methods:** The general idea behind neighborhood methods is to predict the user's preference from a set of similar users or items. The ratings from the user's or item's neighborhoods are computed as a weighted average of the ratings.
- **Deep Learning Methods:** Such methods use advanced neural network architecture to learn high-level, complex, and multi-dimensional relationships from the data. The models are, thus, able to learn deep patterns from the user-item interactions and therefore provide accurate estimation of the values for $\hat{\boldsymbol{R}}_{ui}$.
- **Hybrid Methods:** These combine a wide range of techniques in the hope of exploiting the strengths of each one, thus optimizing the recommendation performance in different scenarios and enhancing the accuracy and reliability of the predictions $\hat{R}_{ui}$.

These methods help in making the prediction better and adjust the prediction of ratings $\hat{\boldsymbol{R}}_{ui}$ as much as possible to the real behavior and preference of users. More details about the types of RSs are in Section 2.1.1, while information about graph theory is in Section 2.1.4.

1.3.4 Evaluation Metrics

The performance of a CFRS is usually evaluated through a variety of metrics, which are designed to measure accuracy and diversity:

- **Prediction Accuracy:** Using metrics such as Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Precision@k and Recall@k. These metrics aim to predict results in terms of the similarity of the predicted ratings $\hat{\boldsymbol{R}}_{ui}$ with actual ratings.

- **Diversity:** Gini Index and Item Coverage are metrics that help assess the variety of items recommended to users, with a wide representation of choices.
- **Others:** More metrics, such as Normalized Discounted Cumulative Gain (NDCG) and F1 Score, take further measurements of other features, such as rank accuracy and the balance between precision and recall.

More details of evaluation metrics can be found in Section 2.1.11.

1.3.5 The General Objective of CFRSs

Given a user set $\mathcal{U}$ and an item set $\mathcal{I}$, for each user $u \in \mathcal{U}$ who interacted with at least one item $i \in \mathcal{I}$, we define the set $\mathcal{I}_u^+ \subseteq \mathcal{I}$ as the set of interacted items, while we define the set $\mathcal{I}_u^- = \mathcal{I} \setminus \mathcal{I}_u^+$ as the set of remaining items. A user interaction with an item includes actions such as rating, clicking, or viewing. The implicit feedback element $\boldsymbol{R}_{u,i}$ is defined as:

$$\boldsymbol{R}_{u,i} = \begin{cases} 1, & \text{if } i \in \mathcal{I}_u^+, \\ 0, & \text{if } i \in \mathcal{I}_u^-. \end{cases}$$

The objective of a CFRS is to predict the value of $\hat{\boldsymbol{R}}_{u,j}$, where $j \in \mathcal{I}_u^-$, and recommend a ranked list of items from $\mathcal{I}_u^-$, associated with the most preferred values of $\hat{\boldsymbol{R}}_{u,j}$. Let us denote the recommended list of items for user u as $\mathcal{L}_u$, which is a subset of $\mathcal{I}_u^-$ ranked based on the predicted preference scores $\hat{\boldsymbol{R}}_{u,j}$. CFRSs aim to learn the predicted preference score $\hat{\boldsymbol{R}}_{u,j} = f(u, j|\Theta)$, where f is an interaction function that maps u and j to a predicted preference score $\hat{\boldsymbol{R}}_{u,j}$. The function f learns the values of the model parameter set Θ through optimizing a selected objective function. In this context, we aim to develop a framework for optimizing CFRSs, driven by the following problem statement.

1.4 Problem Statement

Given a set of users $\mathcal{U}$ and items $\mathcal{I}$, the goal is to develop a CFRS that predicts the preference scores $\hat{\boldsymbol{R}}_{u,j}$ for all $u \in \mathcal{U}$ and $j \in \mathcal{I}_u^-$. The model should then generate a ranked list $\mathcal{L}_u$ of items for each user u, improving the relevance of the recommended items based on their predicted preference scores. At present, CFRS optimization primarily deals with issues of ranking and relevance by improving metrics like recall and Normalized Discounted Cumulative Gain (NDCG). These metrics ensure the retrieval of relevant items and their proper ranking in the recommendation lists $\mathcal{L}_u$ for all $u \in \mathcal{U}$ [18–28].

- **Recall** represents the proportion of relevant items included in $\mathcal{L}_u$, ensuring that users are exposed to enough relevant items in their recommendations.
- **NDCG** measures how effectively the relevant items are ranked within $\mathcal{L}_u$. A higher NDCG score indicates that more important items are positioned closer to the top, leading to improved user satisfaction.

Despite these advances, modern CFRSs still struggle with issues related to **generalization** and **reliability** [6,7,29–34]. These weaknesses necessitate redefining the CFRSs problem to address significant shortcomings in performance. This has led us to redefine the CFRS problem in light of the following challenges:

1. **Diversity and Novelty:** The recommendation list $\mathcal{L}_u$ created from $\mathcal{I}_u^-$ for user u should include a variety of items, including new and unique ones. This helps users explore more options. However, diversity and novelty are often ignored because of the focus on optimizing $\hat{\boldsymbol{R}}_{u,j}$ through recall and NDCG.

2. **Fairness:** Fairness ensures the recommendation list $\mathcal{L}_u$ represents user u preferences from $\mathcal{I}_u^+$ and $\mathcal{I}_u^-$. It prevents overemphasis on popular items in $\boldsymbol{R}$. The aim is to balance the recommendations so that less common preferences are represented fairly while still addressing the user's broader interests.

3. **Long-Tail Recommendations:** Adding long-tail items to recommendations helps to reduce the impact of the sparsity of $\boldsymbol{R}$ and increase the variety in $\mathcal{L}_u$. These items are niche and appeal to minority interests. Traditional optimization metrics like recall and NDCG often overlook them. Including less popular but relevant options makes recommendations more engaging for users with unique preferences and improves the system's ability to serve different interests.

4. **Cold Start:** The cold start problem requires adjusting $\mathcal{L}_u$ for situations where users or items have few or no interactions $\boldsymbol{R}_{u,i}$. Traditional methods for recall and NDCG do not effectively address this issue, so specialized approaches are necessary to integrate new users and items into the system.

5. **Reproducibility Challenges:** It is important for RSs to produce consistent and reliable results. Many systems fail to do so across different environments or when data or configurations slightly change. A robust framework is required to maintain stability and reliability under varying conditions.

6. **Evaluation and Optimization Overfitting:** Evaluation overfitting happens when a RS is overly optimized for a particular dataset or evaluation metric. This can lead to high performance scores that do not reflect the system's actual effectiveness in real-world applications.

   Overfitting during model evaluation and optimization, characterized by:

   $$\text{Error}(\mathcal{F}, \mathcal{G}_{train}) \ll \text{Error}(\mathcal{F}, \mathcal{G}_{test})$$

   where $\mathcal{F}$ is the model being evaluated. The training dataset $\mathcal{G}_{\text{train}}$ is used to fit the model, while $\mathcal{G}_{\text{test}}$ is used to assess its performance on unseen data. RMSE measures the average squared difference between predicted and actual values, making it a useful indicator of error.

   A large difference between training and test RMSE shows that the model fits the training data too closely but does not generalize well. Overfitting results from capturing noise or overly specific patterns in the training data.

Looking at Table1.1, most of the relations between challenges are complementary where addressing a challenge helps in mitigating the other challenges. However, improving Ranking & Relevancy through optimizing NDCG and recall might have an opposite impact on addressing the other challenges. To exemplify, removing unpopular items from the training dataset could improve Recall and NDCG while aggravating the Cold Start problem and the Long-Tail problem. Meanwhile, reproducibility is sensitive to evaluation overfitting as fitting the CFRS testing to a very limited experimental settings might make it very difficult to reproduce the results.

Addressing these challenges requires a comprehensive framework for CFRSs that extends beyond addressing only the ranking and relevancy challenge (improving recall and NDCG) to enhance overall system performance. Further information about the challenges can be found in Section 2.1.7. For further details, Table 1 (In Appendix .1) present a mapping of the proposed solutions and their techniques to the addressed CFRS challenges.

1.5 Awareness Levels for Handling CFRS Challenges

In the context of this section, we classify CFRSs based on the maturity level of the solutions. In other words, each of the proposed CFRSs in this thesis or in litrature have a certain level of maturity in handling the

Table 1.1: Relations Between CFRS Challenges

Characteristic	**Diversity & Novelty**	**Fairness**	**Long-Tail**	**Cold Start**	**Ranking & Relevancy**	**Reproducibility**	**Overfitting**
Diversity & Novelty	-	P	P	P	C	N	P
Fairness	P	-	P	P	C	N	P
Long-Tail	P	P	-	P	C	N	P
Cold Start	P	P	P	-	C	N	P
Ranking & Relevancy	C	C	C	C	-	N	C
Reproducibility	N	N	N	N	N	-	P
Overfitting	P	P	P	P	C	P	-

P: Complementary - Addressing one challenge helps mitigate the other.
C: Competing - Addressing one challenge might exacerbate the other.
N: Neutral - Not necessarily related; No significant direct relationship.

challenges identified in section 1.4. Hence, we try to identify these levels and differentiate the examined CFRSs based on them.

Level 0: Single View Stage. Most of the literature in CFRSs [18–28] belongs to this level as there is no awareness or handling of the challenges identified in section 1.4. For instance, the experiments are tailored to measure NDCG and Recall only, with focus on ranking and relevancy of recommendations.

Level 1: Foundational Experimental Design. At this foundational level, there is an awareness of some of the problems identified in section 1.4. To illustrate, the solution uses ranking and relevancy metrics like Recall and NDCG and focuses on tailoring experiments to measure the CFRS's ability to handle the cold-start problem or the long-tail problem through designing experiments with variable user-item interactions. Furthermore, statistical significance tests show the significance of the results.

Level 2: Metric-Based Evaluation and Improvement. This intermediate level builds on the previous level by utilizing specific metrics (Shannon Entropy, Coverage, Popularity) to evaluate and optimize CFRSs' performance with explicit awareness of the challenges identified in section 1.4. For instance, the experiments could include comparing different CFRSs in terms metrics like Shannon Entropy or Coverage to find the diversity or coverage of the recommended list of items.

Level 3: Comprehensive Quality Standards Compliance. Building on the previous levels, we use different evaluation metrics to measure the CFRS compliance with comprehensive quality standards like ISO 25010. Of course, the design of quality standards is driven by considering challenges in section 1.4.

Level 4: Hybrid Solutions Integration. In certain E-commerce contexts, solutions developed on the previous levels may not solve some complex problems. Therefore, there is could be a need to form hybrid solutions to tackle these challenges, which could be rooted in the challenges described in 1.4. To illustrate, we could use content-based filtering or natural language processing techniques to mitigate the cold-start problem by predicting ratings based on text rather than other rating methods like CFRSs.

1.6 Overview of our Methodological Approaches

This section offers an outline of the methodology used in this study. Adopting software and systems engineering approaches to address the significant challenges associated with CFRS [6,7,29–34], and contribute to the efforts of building a reliable and robust framework based on establishing an analogy with the defined engineering fields. Additionally, as will be later revealed, ISO/IEC 25010:2011 standards [35] are being integrated to provide a comprehensive assessment of the developed CFRSs, allowing for the development of rigorous and high-quality systems.

Adapting Systems Engineering Principles to CFRS: Systems engineering by nature of its interdisciplinary approach enables the end-to-end functional and operability of systems in opposition to requirements. By utilizing means of systems engineering methodologies such as system block diagrams [36–38] on CFRS, this research aids in the visualization and optimization of the embedded network of user and item interactions. By implementing software engineering analogous features to CFRS, specifically through functional analyses [39–41] on recommendation algorithms. Accordingly, this research refines CFRSs toward mitigating CFRSs challenges, such as the cold start problem and data sparsity.

Incorporating Object-Oriented Principles into CFRS Methodology: Leveraging object-oriented programming (OOP) principles to enhance both the structure and performance of an existing methodology. They specify the building blocks of CFRSs by proposing a core model framework that defines three central elements, which are challenges, tools, and experimental design. OOP makes inheritance easy for robust methodologies such as statistical significance tests for model validation and standard metrics like Gini Index, Item Coverage, Shannon Entropy, etc. The implementation of OOP principles allows controlled experiments that keep the system stable and even make it easier—and desirable—to reproduce or compare studies. Last, it can reduce the burdens of customizability and extensibility of CFRSs because it allows for flexible architectures with new embeddings or graph structures. Such an object-oriented design will not only enhance the efficiency of development but also allow easier collaboration and knowledge sharing among researchers, promoting systematic and scalable improvements for CFRS methods.

Incorporating TRIZ for Innovative Problem Solving in CFRSs: The adaptation of TRIZ, a structured and inventive problem-solving methodology originating from engineering [42, 43], to the domain of CFRSs has allowed developing creative solutions for technical contradictions, restricting the diversity and accuracy of recommendations. Through the incorporation of TRIZ principles, this thesis has introduced innovative strategies to optimize CFRSs by promoting adaptability and user-centered performance, allowing RSs to improve with changes to user preferences and item ratings.

Holistic Application of Engineering Standards to CFRSs: Using engineering principles [42, 43] along with a structured approach that aligns with industry standards like ISO 25010, improving the design and evaluation of CFRSs. This ensures the systems meet key quality requirements such as functionality, reliability, and usability. Avoiding ad hoc methods, this approach strengthens the system's structure and improves its ability to handle diverse and complex operational needs. The result is recommendations that are fair, personalized, and meet individual user needs while supporting long-term engagement goals and following industry standards.

1.7 Research Objectives

In order to achieve the above goals, this research will pursue the following specific objectives:

1. To conduct a detailed analysis of the limitations and potential areas of improvement of the existing CFRSs.

2. Develop a comprehensive recommendation integration system using such tools as the System Block Diagram for architectural presentation, Functional Analysis, and TRIZ principles for creative problem-solving.

3. Evaluate the developed CFRs based on the criteria proposed by ISO/IEC 25010:2011 with a specific focus on aspects of reliability, usability, performance efficiency, and privacy to ensure that the developed systems meet the standards.

4. Develop cutting-edge recommender solutions based on the integration of these methodologies for E-commerce platforms. For illustration, these solutions will then be tested empirically to confirm if they have improved recommendation accuracy, increasing user engagement and satisfaction.

5. Disseminate the results of the research and the presented methodologies through scholarly publications and follow industrial standards in developing CFRSs solutions, for instance evaluating solutions using ISO/IEC 25010:2011 standards.

These specific objectives have been developed with the intent to cover all relevant issues that surround the core function of CFRSs. Hence, the current study proposes a complex solution to the problem and a comprehensive evaluation of the developed systems. Upon completion, the research will be able to significantly impact the current state of the art in the realm of CFRSs technologies and provide a solid ground for future exploration.

1.8 List of Publications

The following represents a list of publications:

Published Papers:

1. Alshareet, O., Itradat, A., Doush, I.A. and Quttoum, A., 2018. Incorporation of ISO 25010 with machine learning to develop a novel quality in use prediction system (QiUPS). *International Journal of System Assurance Engineering and Management*, Springer, pp.344-353.

2. Alshareet, O. and Awasthi, A., 2023. Enhancing E-commerce recommendations with a novel scale-aware spectral graph wavelets framework. *International Journal of Data Science and Analytics*, pp.1-14, Springer Switzerland.

3. Quttoum, A.N., Alsarhan, A., Moh'd, A., Alshareet, O., Nawaf, S., Khasawneh, F., Aljaidi, M., Alshammari, M. and Awasthi, A., 2023. ABLA: Application-Based Load-Balanced Approach for Adaptive Mapping of Datacenter Networks. *Electronics*, 12(17), p.3689.

4. Alshareet, O. and Hamza, A.B., 2024. Adaptive spectral graph wavelets for collaborative filtering. *Pattern Analysis and Applications*, 27(10), Springer London.

5. Alshareet, O. and Awasthi, A., 2024. Enhancing E-Commerce Complaint Resolution: Adherence to ISO 10001:2018 with LSTM-Based Sentiment Analysis. *The CORS Annual Conference 2024*, London, Ontario, Canada.

Accepted Papers with Revision:

1. Alshareet, O. and Awasthi, A., Light Spectral-based Graph Convolutional Networks: Optimizing Ranking and Long-tail Awareness for E-commerce Recommendation Systems. *Knowledge and Information Systems*, Springer. Status: Decision is major revision. (Solution 2 chapter)

2. Alshareet, O. and Awasthi, A., A Novel Framework for Integrating Blockchain-Driven Federated Learning with Neural Networks in E-Commerce. Status: under peer review.

Papers Under Peer-review:

1. Alshareet, O. and Awasthi, A., Federated Learning and Big Data Analytics: A Novel Framework for Crafting Codes of Conduct for E-commerce Supply Chains. Status: under peer review.

Papers In-progress:

1. Alshareet, O. and Awasthi, A., Leveraging ISO/IEC 25010:2011 and Temporal Aware Collaborative Filtering for Improved E-Commerce Recommendations.

1.9 Research Contributions

This section provides an elaboration on the main contributions of the study, describing the new approaches and methodologies applied for the development of improved CFRSs. The following subsections then present the contributions and implications of the main novelties for the two domains—namely, the theoretical and practical applications of the field.

Development of a Novel Hybrid Framework: One of the research contributions is represented in merging System Block Diagrams, Functional Analysis, and TRIZ methodologies with CFRSs. A merger of all three methods in addressing the previous gaps while facilitating the improvement in both structural and functional sides of the CFRSs. In the end, this innovation may lead to better solutions in terms of robustness and novelty in E-commerce platforms.

Novel Integration of ISO/IEC 25010:2011 Standards in CFRSs: The second contribution lies in the integration of ISO/IEC 25010:2011 standards with existing collaborative CFRSs' performance evaluation. Utilizing ISO/IEC 25010:2011 standards in the existing evaluation and enhancement framework allows for a specific high-quality benchmark to be followed. As a result, the existing inefficiencies can be identified and removed in a systematic manner that makes data scientists and developers aware of the quality of the developed CFRSs. Furthermore, guiding the development of CFRSs with ISO/IEC 25010:2011 standards enables commercializing CFRSs applications.

Empirical Validation in Modeled Real-world E-commerce Scenarios: The proposed methodologies are backed by the use of real-world datasets to model E-commerce scenarios. This method allows for detailed testing and validation in conditions similar to actual situations, adding value to the research. These modeled scenarios show how the methodologies can increase user satisfaction and engagement, linking the theoretical parts of the research to practical uses and supporting preparation for real-world implementation.

Comprehensive Analysis and Enhancement of Existing Systems: The research conducts a detailed analysis of current CFRSs, identifying inefficiencies and areas needing improvement. This critical examination helps to deepen the understanding of necessary enhancements and how they can be effectively implemented using the newly introduced hybrid framework.

Implementation and Evaluation of Advanced CFRSs Presented in Peer-Reviewed Papers: Additionally, this research clearly defines and empirically validates advanced CFRSs, which have been documented and disseminated through peer-reviewed publications. These methodologies, integrated with real-world datasets and simulated scenarios, demonstrate significant enhancements in recommendation accuracy and user engagement. The publication of these findings not only proves their practical utility but also significantly contributes to advancing the field of CFRSs, demonstrating their academic value.

1.10 Thesis Blueprint

In this section, we outline the structure and content of the forthcoming chapters. Each chapter is designed to build upon the knowledge established in the previous ones, methodically guiding the reader through the intricate details and innovative methodologies applied within the field of CFRSs.

Chapter 1: Introduction

The first chapter provides a comprehensive introduction to CFRSs, emphasizing their significance in E-commerce. It explores collaborative filtering mechanisms, detailing approaches and solution strategies,

and discusses various evaluation metrics and objectives for CFRSs. The chapter culminates with research Contributions, aims, objectives, and other related sections.

Chapter 2: Conceptual and Literature Foundations

This chapter establishes the conceptual and research foundations for CFRSs. It explores foundational concepts, including collaborative, content-based, and hybrid filtering, along with graph-based models and neural networks. Challenges such as the cold start problem, long tail effects, and diversity are addressed alongside advancements in algorithms and evaluation methods. Key topics like evaluation metrics, software quality, and datasets are also discussed to provide a comprehensive overview of the field.

Chapter 3: A Methodological Framework for Optimizing CFRSs

This chapter introduces a methodological framework aimed at optimizing CFRSs. It outlines the strategic implementation of systems block diagrams and functional analysis in the development of CFRSs. The application of the Theory of Inventive Problem Solving (TRIZ) in addressing contradictions within CFRS algorithms is highlighted, followed by related work and methodological details.

Chapter 4: Solution 1 - An ISO-based Temporal Graph Collaborative Filtering Recommender System

This chapter presents the first solution involving a temporal graph collaborative filtering (TGCF) system based on ISO standards. It discusses the application of graph convolutional networks (GCNs), detailing the matrix operations involved, and explores various performance metrics through experimental analyses to validate the system's efficacy.

Chapter 5: Solution 2 - Hybrid Graph Convolutional Networks for Optimizing Ranking and Long-tail Awareness of E-commerce Recommendation Systems

This chapter discusses the integration of hybrid graph convolutional networks (GCNs) to enhance E-commerce RSs. It explores various methodologies, including graph Fourier transforms and spectral collaborative filtering, and evaluates these approaches through detailed experiments.

Chapter 6: Solution 3 - Enhancing E-Commerce Recommendations with a Novel Scale-Aware Spectral Graph Wavelets Framework

This chapter introduces a novel framework utilizing scale-aware spectral graph wavelets to enhance recommendation accuracy and efficiency in E-commerce systems. It covers the design principles, eigenvalue filtering techniques, and the implementation of a neural multi-layer model, assessing its performance through comprehensive experimental analysis.

Chapter 7: Solution 4 - Adaptive Spectral Graph Wavelets for CFRSs

This chapter explores adaptive spectral graph wavelets in collaborative filtering. It details the methodological approach, from preliminary concepts to proposed methods, and assesses the performance across various user interaction levels and recommendation scenarios through extensive testing.

Chapter 8: Conclusions and Future Works

The final chapter summarizes the main contributions to knowledge, discussing the critical examination of quality attributes in CFRSs and the innovative application of engineering methodologies. It outlines the limitations of current studies and suggests directions for future research and system innovations.

Chapter 2

Conceptual and Literature Foundations

This chapter presents the basic ideas and research for CFRSs. It has two main parts, Preliminaries and Literature Review.

The Preliminaries section introduces basic concepts like collaborative filtering, content-based filtering, and hybrid systems. It also includes introductory concepts like graph-based models and neural networks. There are also topics about evaluation, software quality, and datasets.

The Literature Review part explains the main problems in the field and related research. It covers issues like the cold start problem, long tail effects, and diversity. It also looks at progress in algorithms and evaluation methods. This chapter attempts to make the thesis more accessible to readers.

2.1 Preliminaries

The Preliminaries section sets up the required understanding to take forward discussions and contributions of this research. It includes a detailed study on types and techniques of RSs with illustrative examples to clarify each type of technique. Further, briefly clarifying the general introduction to graph theory, this section covers the necessary aspects associated with graph theory, which come into play with collaborative filtering and, subsequently, the application of the graph-based models. Further, this includes the challenges existing in the current scenario with collaborative filtering and the application of graph neural networks and graph convolutional networks for their improvement. Respective software quality standards as ISO/IEC 25010:2011 cover evaluation metrics and benchmark datasets, as well as measure the performance of the CFRSs.

2.1.1 Types and Techniques of Recommender Systems

RSs have been defined as algorithms that support the user through recommending items, movies, books, products, and so forth. RSs can be broadly divided into three types: collaborative filtering, content-based filtering, and hybrid methods.

Collaborative Filtering

Collaborative filtering is an approach that recommends items that the user might like. In a broader sense, it is a process used for the purpose of filtering and analyzing a huge amount of information about user behaviors, activities, or preferences toward predicting what users will like based on similarity with other users. There are two kinds of collaborative filtering techniques:

- **User-based Collaborative Filtering:** This technique makes recommendations to users based on the similarity of the users. If User A shows interests similar to those of User B, the system recommends items that User B likes and User A has not yet seen.
- **Item-based Collaborative Filtering:** The system recommends items that are similar to the ones a user has liked in the past. In general, it is more stable and scalable than user-based filtering.

Content-Based Filtering

Content-based filtering makes recommendations based on the attributes of items and a profile of the user. For example, it is the determination of the system to notice that a user likes to watch action movies, and, subsequently, it recommends films under the category of "action."

Hybrid Methods

Hybrid models take the best features of the two methods: collaborative and content-based filtering. The hybrid methods therefore offer more effective recommendations for a specific user's tastes, better and more precise recommendations.

2.1.2 Example Ratings Table

Consider the following example of a ratings table where users rate movies on a scale from 1 to 5:

User / Movie	Movie 1	Movie 2	Movie 3	Movie 4
User 1	4	?	3	5
User 2	3	3	?	4
User 3	?	4	2	3
User 4	2	2	3	?

Next, we calculate the unknown values using different types of RSs:

Collaborative Filtering Example

User-based CF: To predict User 1's rating for Movie 2:

- Identify users with similar rating patterns to User 1.
- Calculate the average rating for Movie 2 from similar users (User 2 and User 4): $(3 + 2)/2 = 2.5$, rounded to 3.

Item-based CF: To predict User 2's rating for Movie 3:

- Consider items similar to Movie 3 that User 2 has rated.
- Calculate the average of similar ratings: $(3 + 4)/2 = 3.5$, rounded to 4.

Content-Based Filtering Example

To predict User 3's rating for Movie 1, based on other action movies User 3 rated:

- Assume Movie 1 is similar to Movies 2 and 4.
- Calculate the average rating for similar movies: $(4 + 3)/2 = 3.5$, rounded to 4.

Hybrid Methods Example

To predict User 4's rating for Movie 4:

- Use both user-based CF and content-based predictions.
- User-based CF suggests 4 (average of ratings by User 1, User 2, and User 3).
- Content-based prediction suggests 3.
- Combine these to get $(4 + 3)/2 = 3.5$, rounded to 4.

2.1.3 Rationale for Choosing CFRSs

CFRS is selected as the primary RS methodology for this PhD thesis due to its distinct advantages and alignment with the thesis's goals and challenges, as demonstrated in the comparative table of RSs [44–46] (Table 2.1). The key reasons for choosing CFRS include:

- **Advanced Personalization through User-Item Interactions:** One of the great strengths of CF is in its capacity to derive user preferences from user-item interactions, allowing it to provide far superior personalization compared to knowledge-based or demographic-based techniques, which makes it highly desirable for its capacity to adapt to individual user preferences. Table 2.1 shows the effectiveness of CF to adapt to individual user preferences.
- **Minimal Data Requirements:** CFRSs only require small amounts of initial data, generally just user, item, and rating information. This is unlike other systems that have to store either item attributes in detail in Content-Based systems or user profiles in **volumes** in Demographic systems. This reduces the overhead of setup and maintenance of the data, making the technique fit for applications in dynamic environments, thus adaptable with necessity.
- **Research Opportunities in Addressing Data Sparsity and Cold Start:** CFRS techniques are based on improving the quality of the system but are often vitiated by problems of data sparsity and cold starts. These happen to be two important gaps in the research that this thesis tries to fulfill. Accordingly, the research will enhance system performance through access to advanced techniques, like matrix factorization, and devising new metrics to work on some of the challenges presented herein. This approach will use the data recorded from the user's interaction to solve these problems and therefore result in better system quality.
- **Optimal Scalability for Research Context:** CF offers medium scalability, providing a framework balanced in effectiveness and potential growth of the system. This kind of scalability suffices for the research context to be used, where the emphasis is on user experience and quality of personalized recommendations, and not on large data handling.

Overall, the adoption of CFRS is justified by its ability to deliver superior personalization and the significant opportunities it presents to address critical research gaps associated with data sparsity and the cold start problem. These gaps align closely with the thesis's objectives to advance knowledge in RS efficacy and user satisfaction, solidifying the rationale for selecting CF as the foundational methodology.

2.1.4 Concepts in Graph Theory

In this subsection, we will cover basic topics, which should help get a clearer view of sophisticated concepts within graph-structured data. We define mathematical formulations essential to understand the basic structure and dynamics of the graphs in sufficient manner to understand the thesis.

Graphs and Bipartite Graphs: Graphs, typically written as $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of vertices and $\mathcal{E}$ is the set of edges, play a crucial role in the context of modeling relations between entities, both in mathematics and computer science. It brings manifold benefits in E-commerce for characterizing and analyzing complex interactions. Bipartite graphs, $\mathcal{G}(\mathcal{U}, \mathcal{I}, \mathcal{E})$, where both $\mathcal{U}$ and $\mathcal{I}$ are sets of vertices that are disjoint and correspond to the users and items, respectively, and $\mathcal{E}$ is denoting the edges between these sets, play extremely important roles in E-commerce and recommendation systems. An edge $(u, i) \in \mathcal{E}$ means a relationship between user $u \in \mathcal{U}$ and item $i \in \mathcal{I}$, which is the basis for the utility of this graph.

The following applications illustrate how bipartite graphs have been utilized within E-commerce, and these have proved an extensive ability to model different kinds of interactions within this area.

Table 2.1: Comparison of Different Types of Recommender Systems

Type	Method[1]	Personalization[2]	Data Sparsity[3]	Cold Start[4]	Scalability[5]
Collaborative Filtering	User-Item Interactions	High	High	High	Medium
Content-Based	Item Features	Medium	Medium	Medium	High
Hybrid	Combination of Methods	High	Low	Low	Medium
Knowledge-Based	Domain Knowledge	Low	N/A	N/A	High
Demographic	User Demographics	Low	N/A	N/A	High
Utility-Based	Utility Functions	Medium	N/A	N/A	Medium

[1] Method refers to the fundamental approach used by the RS.
[2] Personalization level: High, Medium, Low.
[3] Data Sparsity issue: High (significant issues), Medium (some issues), Low (minimal issues).
[4] Cold Start problem severity: High (significant difficulty), Medium (moderate difficulty), Low (minimal difficulty).
[5] Scalability: High (excellent scalability), Medium (good scalability), Low (poor scalability).
[6] Method: More details about the criteria behind this table is available in Appendix .2.

Rating and Review Analysis Graphs: A specialized kind of bipartite graph, $\mathcal{G}(\mathcal{U}, \mathcal{R}, \mathcal{E})$, where $\mathcal{R}$ is the set of reviews and $(u, r) \in \mathcal{E}$ is used to represent the fact that user u has reviewed item r. The graph can be instrumental in analyzing the consumer feedback patterns, thus helping to fine-tune the recommendation systems, further drawing insights for the betterment of the product from reviews given by the customers.

These applications represent how bipartite graphs can function in E-commerce. They serve as a good example of a flexible scheme within complex networks of interactions between users and items can be modeled and analyzed, thus actually inciting development of very advanced tools and systems engineered to get the most out of customer experiences in terms of recommendations.

Representation Learning on Graphs: Representation learning on graphs aims to simplify the complex relationships between nodes into low-dimensional forms. This helps in tasks like classification, prediction, and clustering and captures the main features of nodes, edges, or subgraphs.

Graph Embedding Techniques and Latent Factors: There are a variety of applications of the graph embedding techniques in the area of enabling translation from highly complicated, high-dimensional relationships in a graph to much more informative and easy-to-use low-dimensional descriptions. In this sense, graph embedding techniques can encompass a family of methodologies to draw more utility and insight from graph data.

Factorization-Based Approaches: These approaches factorize matrices associated with the graph—like the adjacency matrix $\mathbf{A}$ or Laplacian matrix $\mathbf{L}$—to obtain the embeddings. One of the most common ways is known as spectral factorization, by using the eigendecomposition of the Laplacian matrix, defined as $\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{\mathrm{T}}$, from whose eigenvectors $\mathbf{U}$, significant structural patterns in the graph are inferred. These methods sample walks from random processes over the graph, after which they are trained to reproduce co-occurrences of nodes within these walks in low-dimensional space, thereby capturing the topology and community structure of the graph.

Deep Learning-Based Methods: By using deep learning frameworks on the graph, with the help of Graph Convolutional Networks (GCNs) and Graph Attention Networks (GATs), one can learn these embeddings by explicitly making use of the graph structure. For instance, mathematically, a GCN layer operates like $\mathbf{H}^{(l+1)} = \sigma(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}^{(l)}\mathbf{W}^{(l)})$, where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ is the adjacency matrix with added self-loops, and $\tilde{\mathbf{D}}$ is the degree matrix of $\tilde{\mathbf{A}}$.

2.1.5 Expanding Collaborative Filtering with Graph-Based Models

The merger of graph theory and representation learning with CF represents a significant shift in the traditional conception and implementation of recommendation systems. This expanded discussion explores the depths of graph-based collaborative filtering, highlighting its mathematical underpinnings and the broadened scope of its applications.

Mathematical Modeling of Graph-Based CF: In the graph-based CF framework, user-item interactions are modeled as a bipartite graph $\mathcal{G} = (\mathcal{U}, \mathcal{I}, \mathcal{E})$, with $\mathcal{U}$ and $\mathcal{I}$ representing user and item sets, respectively, and $\mathcal{E}$ indicating interactions.

Adjacency Matrix. The bipartite graph $\mathcal{B}$ has an adjacency matrix $\boldsymbol{A}$ that can be defined as:

$$\boldsymbol{A}_{N\times N} = \begin{bmatrix} 0 & \boldsymbol{R} \\ \boldsymbol{R}^{\intercal} & 0 \end{bmatrix}, \tag{2.1}$$

where $N \times N$ are dimensions for $\boldsymbol{A}$, $\boldsymbol{R}$ is an implicit feedback matrix with dimensions $|\mathcal{U}| \times |\mathcal{I}|$, $\boldsymbol{R}_{u,i} = 1$ if there is a relation between $u \in \mathcal{U}$ and $i \in \mathcal{I}$, otherwise $\boldsymbol{R}_{u,i} = 0$.

Laplacian Matrix. The Laplacian matrix is defined as $\boldsymbol{L} = \boldsymbol{I} - \boldsymbol{D}^{-1}\boldsymbol{A}$, where $\boldsymbol{I}$ is the identity matrix, and $\boldsymbol{D}^{-1}$ is the inverse of the diagonal degree matrix. The matrix $\boldsymbol{L}$ possesses a set of eigenvectors $\boldsymbol{U} = (\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_N)$, and a corresponding set of eigenvalues $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_N)$, which is a diagonal matrix composed of the eigenvalues.

Normalized Laplacian Matrix. the following equations represent the normalized Laplacian:

$$\boldsymbol{L}^{\eta} = \boldsymbol{U}\Lambda\boldsymbol{U}^{\boldsymbol{T}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{L}\mathbf{D}^{-\frac{1}{2}} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}} \tag{2.2}$$

Where $\mathbf{L}^{\eta}$ has a set of orthogonal eigenvectors $\mathbf{U} = (\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_N)$ and is accompanied by a set of eigenvalues $\widetilde{\boldsymbol{\Lambda}} = \text{diag}([\tilde{\lambda_1}, \tilde{\lambda_2}, \ldots, \tilde{\lambda}_N])$. Thus, it is often necessary to find its eigenvectors $\mathbf{U}$ and eigenvalues $\widetilde{\boldsymbol{\Lambda}}$. However, full computation can be resource-intensive, especially for large matrices.

Leveraging Latent Factors in CF through Embeddings: Embedding users and items into a shared latent space aims to accurately predict their interactions. This involves minimizing the discrepancy between predicted $\hat{\mathbf{A}}$ and observed interactions $\mathbf{A}$, typically employing loss functions like mean squared error (MSE) or binary cross-entropy.

Deep Learning Enhancements in Graph-Based CF: Techniques such as GCNs and GATs enhance embedding accuracy by integrating graph structure into the learning algorithm. For example, GCN embeddings update as:

$$\mathbf{H}^{(l+1)} = \sigma\left(\hat{\mathbf{D}}^{-\frac{1}{2}}\hat{\mathbf{A}}\hat{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}^{(l)}\mathbf{W}^{(l)}\right),$$

where $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ adds self-connections, $\hat{\mathbf{D}}$ is $\hat{\mathbf{A}}$'s degree matrix, $\mathbf{H}^{(l)}$ is the l^{th} layer's embedding matrix, and $\mathbf{W}^{(l)}$ is the trainable weight matrix.

Wider Use Cases: Graph-based collaborative filtering will provide a path for many diversified recommendation applications, including content-based, cross-domain, and social recommendation systems. Further

integration of the social network graphs of the users will make the recommendations more personal and more precise.

By merging collaborative filtering with graph theory and state-of-the-art techniques in representation learning, we gain a powerful tool to build more complex, adaptive, and engaging recommendation systems. At the core, this makes the present approach also quite comprehensive in deepening the computational understanding of user-item interactions and widening the horizons for creative recommendation strategies in order to be responsive to varied needs and preferences of users in the digital world.

2.1.6 Introducing GFT

In this section, we introduce prerequisites for Graph Fourier Transform (GFT) .

Graph Convolution and GFT. GFT considers the eigenvectors of normalized Laplacian matrix as a set of bases. The GFT of a signal $\boldsymbol{x} \in \mathbb{R}^N$ on a graph $\mathcal{G}$ as $\hat{\boldsymbol{x}} = \boldsymbol{U}^\intercal \boldsymbol{x}$, whereas the inverse of GFT as $\boldsymbol{x} = \boldsymbol{U}\hat{\boldsymbol{x}}$ [47]. Additionally, the graph $\mathcal{G}$ operator $*_\mathcal{G}$ is defined as follows:

$$\boldsymbol{x} *_\mathcal{G} \boldsymbol{y} = \boldsymbol{U}((\boldsymbol{U}^\intercal \boldsymbol{y}) \odot (\boldsymbol{U}^\intercal \boldsymbol{x})) \tag{2.3}$$

The symbol $\odot$ indicates element-wise Hadamard product while the symbol $\boldsymbol{y}$ presents the convolution kernel. The Hadamard product is transformed into a matrix multiplication [48] by substituting the vector $\boldsymbol{U}^\intercal \boldsymbol{y}$ with the diagonal matrix $g_\varphi(\Lambda)$, where $g_\varphi\ (\Lambda) = diag\ ([\varphi_0\lambda_0,\ \varphi_1\lambda_1,\ \varphi_{N-1}\lambda_{N-1}])$. Therefore, the Equation 2.3 is represented as follows:

$$\boldsymbol{x} *_\mathcal{G} \boldsymbol{y} = \boldsymbol{U} g_\varphi(\Lambda)\boldsymbol{U}^\intercal \boldsymbol{x}. \tag{2.4}$$

In this research context, we use two types of graph signals for $\mathcal{B}$, which are the user vector $\boldsymbol{x}^u \in \mathbb{R}^{|U|\times 1}$ and the item vector $\boldsymbol{x}^i \in \mathbb{R}^{|I|\times 1}$. Consequently, we replace signal $\boldsymbol{x}$ in Equation 2.4 with a vector representing both signals $\begin{bmatrix}\boldsymbol{x}^u \\ \boldsymbol{x}^i\end{bmatrix}$, and the bipartite graph convolution operator $*_\mathcal{G}$ is defined as follows:

$$\begin{bmatrix}\boldsymbol{x}^u \\ \boldsymbol{x}^i\end{bmatrix} *_\mathcal{G} \boldsymbol{y} = \boldsymbol{U} g_\varphi(\Lambda)\boldsymbol{U}^\intercal \begin{bmatrix}\boldsymbol{x}^u \\ \boldsymbol{x}^i\end{bmatrix} \tag{2.5}$$

In Equation 2.5 the filter $g_\varphi(\Lambda)$ learns parameters φ from the spectral domain of a bipartite graph $\mathcal{B}$.

We define the GFT of a bipartite graph as $\begin{bmatrix}\hat{\boldsymbol{x}}^i \\ \hat{\boldsymbol{x}}^u\end{bmatrix} = \boldsymbol{U}^\intercal \begin{bmatrix}\boldsymbol{x}^u \\ \boldsymbol{x}^i\end{bmatrix}$. In Equation 2.5, a convolution filter $g_\varphi\ (\Lambda)$ is placed on a the transformed spectral *graph signal* $\begin{bmatrix}\hat{\boldsymbol{x}}^u \\ \hat{\boldsymbol{x}}^i\end{bmatrix}$, where each value of the learned parameters φ allows boosting or shrinking the corresponding signal frequency.

GFT limitations. research literature shows performance issues to operations associated with GFT. For example, the multiplication between the highly-dense transformation matrix $\boldsymbol{U}^\intercal$ and the signal $\boldsymbol{x}$ generates a dense projected signal, which is inefficient computationally [48]. Table 2.2 shows that the GFT basis matrix $\boldsymbol{U}$ is highly dense for all datasets, which makes it less efficient computationally. Also, in GFT context, matrices $\boldsymbol{U}$ and $\boldsymbol{U}^\intercal$ don't describe the local topology of a projected node [48], which reduces the efficiency of graph-based semi-supervised learning.

Furthermore, since GFT is not localized in a node local domain, it becomes difficult to restrict the node neighborhood during the convolution process. This tends to make GFT typically restrict neighborhoods using the shortest path distance approach. Several methods have been developed for approximating the eigenvalues and eigenvectors of the Laplacian [49]. The accuracy of spectral methods used to estimate eigenvalues and eigenvectors of matrices often depends on the matrix structure and the method used. It has been reported in the literature that spectral-based methods often perform better in estimating the eigenvalues and eigenvectors

Table 2.2: Sparsity of GFT basis matrix.

Dataset	$\boldsymbol{U}$ Sparsity
ACS	1.9%
MI	0.72%
Gowalla	0.22%

of bipartite graphs than the Chebyshev polynomial-based methods since the positive and negative eigenvalues usually differ distinctly [49].

2.1.7 Challenges in CFRSs

Data Sparsity: Data sparsity is an endemic challenge in bipartite graphs $\mathcal{G}(\mathcal{U}, \mathcal{I}, \mathcal{E})$ representing user-item interactions. Most user-item interactions remain unobserved or unknown. This results in a sparse interaction matrix R, which reduces the ability of CFRS's algorithms to predict future interactions from existing ones.

$$\text{Sparsity}(\mathcal{G}) = 1 - \frac{|\mathcal{E}|}{|\mathcal{U}| \times |\mathcal{I}|}$$

This formula quantifies the level of sparsity, with a higher value indicating fewer observed interactions relative to the total possible interactions, making prediction tasks more challenging.

Cold Start Problem: When users or items are new, there is no historical interaction data available to help the model predict $\hat{\boldsymbol{R}}_{u,j}$. This reduces the complexity of the bipartite graph $\mathcal{G}$ and makes it harder to identify links between users and items. This leads to lower quality and less relevant recommendations. Solutions such as demographic-based or content-based recommendations can help address this. Figure 2.1 shows that the absence of a connection between item $i1$ and user $u1$ demonstrates the cold-start problem. In this situation, the CFRS model finds it difficult to recommend items for $u1$. This problem happens when the model cannot suggest popular or suitable items to users who lack sufficient historical data. It usually impacts new users or those with very limited activity, potentially pushing them away from the platform [50].

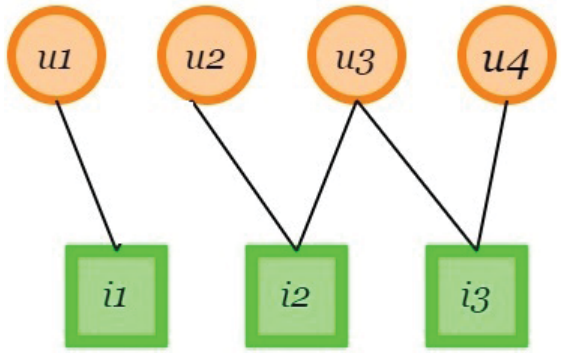


Figure 2.1: A Bipartite graph for representing the relation between users and items.

Long-Tail Problem: The long-tail problem (Figure 2.2) is very common in CFRSs. This is because most of the recommendations are based on historical item data. CFRSs tend to exhibit high favoritism towards items with rich historical interaction and often recommend popular items or items which are always in demand among the users. Such popular items are in the "head" of the distribution, and hence, are a lot more recommended because these items have been shown relevant to a large number of users. Though approaches that try to explore and leverage properties of items or user demographics might seem a way to navigate around this

issue, such methods only enhance the ability of CFRSs to explore and recommend from the long tail. Since a minority of items will gather the majority of the interactions, it will be hard to effectively prioritize items from $\mathcal{I}_u^-$ in a way that diversifies recommendations. The model's inherent bias towards popular items might shadow potentially relevant items in the long tail, thus limiting the possibility to diversify recommendations and effectively explore the item space [51].

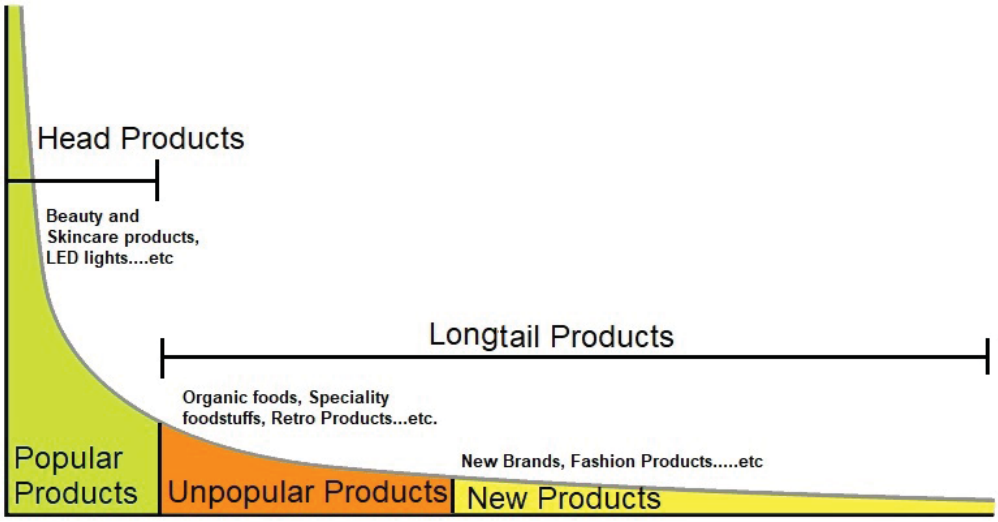


Figure 2.2: Illustration of the long-tail problem, highlighting the disparity in interaction distribution among items, with a minority (head) accumulating the majority of interactions and a majority (long tail) accruing sparse interactions.

Each challenge necessitates nuanced strategies and considerations, ensuring that the CFRSs not only predicts user-item interactions accurately but also navigates the complexities and ethical considerations inherent in real-world applications.

2.1.8 Graph Neural Networks (GNN) in CFRSs

Detailed Overview of Graph Neural Networks (GNN) with Mathematical Framework Graph Neural Networks (GNNs) effectively process graph-structured data, where nodes represent entities and edges define the connections or relationships between these entities. Unlike traditional neural network architectures, GNNs are specifically designed to leverage the inherent connectivity structure of graphs, enabling them to capture both node features and relational information. The operations within a GNN can be mathematically represented similarly to those in traditional Multi-Layer Perceptrons (MLPs), but with the inclusion of the adjacency matrix to integrate neighborhood features. This can be formalized as follows:

$$\mathbf{H}^{(l)} = \sigma\left(\mathbf{W}^{(l)}\left(\mathbf{A}\mathbf{H}^{(l-1)}\right) + \mathbf{b}^{(l)}\right)$$

where:

- $\mathbf{H}^{(l)}$ denotes the matrix of node features at the l^{th} layer, with $\mathbf{H}^{(0)} = \mathbf{X}$ representing the initial features of the nodes.
- $\mathbf{W}^{(l)}$ and $\mathbf{b}^{(l)}$ are the weight matrix and bias vector for the l^{th} layer, respectively.
- σ is the non-linear activation function applied element-wise.
- $\mathbf{A}$ is the adjacency matrix, which encodes the graph structure and is used to propagate features between connected nodes effectively handling the aggregation of neighborhood features.

- $\mathbf{H}^{(l-1)}$ is the output from the previous layer $(l-1)$, serving as the input to the current layer.

This design leverages the characteristics of the nodes and their connections represented by the adjacency matrix so that the network can learn complex dependencies within the graph. Applying the weights and biases at each layer helps the GNN in refining and adapting the understanding of the data while producing rich and informative node embeddings capturing both the attributes of the nodes and the overall graph structure. GNNs have been shown to be highly effective in a broad, exhaustive range of applications where data is naturally structured into graphs. GNNs can, by their very nature of directly accommodating relational information, be a powerful method in extracting deep insights from complex datasets.

2.1.9 Graph Convolutional Networks (GCN) in CFRSs

Detailed Overview of Graph Convolutional Networks (GCN) with Network Architecture Graph Convolutional Networks are one of the most important breakthroughs in deep learning that were designed to work with graph-structured data. They elaborate on the notion of a convolution operation, which constitutes a fundamental tool for the processing of Euclidean data in the graph domain. In the graph, the entities are called nodes and the relation between them is denoted by edges, making the topology of nodes. The matrix of the node features X and the adjacency matrix A of the graph on which the graph convolutional layer is to operate get aggregated using features from the adjacent nodes. This way, the GCN can access both local and global structural details. This layer can be described in the following formula:

$$H^{(l+1)} = \sigma(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}H^{(l)}W^{(l)})$$

where:

- $H^{(l)}$ denotes the matrix of node features at the l^{th} layer, with $H^{(0)} = X$, the initial input feature matrix.
- $\hat{A} = A + I_N$ is the adjacency matrix A augmented with self-connections I_N (the identity matrix), enabling nodes to incorporate their own features during aggregation.
- $\hat{D}$ is the diagonal degree matrix for $\hat{A}$, where each diagonal element $\hat{D}_{ii} = \sum_j \hat{A}_{ij}$.
- $W^{(l)}$ represents the weight matrix for the l^{th} layer, a parameter learned during the training process.
- $\sigma(\cdot)$ is a non-linear activation function, such as the ReLU function defined as $\sigma(x) = \max(0, x)$.

This equation shows how graph convolution updates the representation of each node through aggregating attributes from its neighbors, with the normalization factor $\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}$, which scales the features of neighboring nodes. This normalization process is important to prevent nodes with high degrees from having the most significant influence on training. The remaining layers in GCNs allow for continuous updating of node embeddings. Each layer expands the level of neighborhood information incorporated into the embeddings of nodes. This multilayer approach enables GCNs to learn significant patterns of node connectivity and features, leading to embeddings that reflect both local and global properties of the graph structure. Leveraging the efficacy of GCNs in processing extensive graph-based datasets makes them extremely useful for different applications, such as node classification, link prediction, and studying intricate relationship networks in CFRSs. The generalization of convolutional operations to adapt them to non-Euclidean data greatly expands the scope of their potential use within neural networks and promises breakthroughs in multiple knowledge-intensive fields.

2.1.10 Comprehensive Overview of Software Quality Standards

Introduction to ISO/IEC 25010:2011

The International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) develop standards that serve as global references for products, services, and systems. These standards aim to ensure quality, safety, and efficiency in various fields. ISO/IEC 25010:2011 [35] is part of the Systems and Software Quality Requirements and Evaluation (SQuaRE) group. It provides a model for evaluating the quality of software products. This standard builds on ISO/IEC 9126, which divided software quality into two main categories: Product Quality and Quality in Use. ISO/IEC 25010:2011 further splits Product Quality into eight characteristics that help assess whether the software meets its intended performance requirements.

Product Quality Product Quality in ISO/IEC 25010:2011 [35] is further divided into eight characteristics, which are essential for assessing the software's capability to maintain its performance in accordance with specified requirements. These characteristics are:

1. **Functional Suitability:** This is regarded as one of the key characteristics and establishes the presence of software that has the necessary functions that meet stated and purported needs under specific conditions. This is done with a view of aligning user demands to the software and the results it is expected to deliver. This is further subdivided into completeness (availability of all necessary functions), correctness (supplying accurate outputs with necessary precision), and appropriateness (usefulness of functions for particular tasks). This relates to the satisfaction of the user's orders and customers' needs generally.

2. **Performance Efficiency:** This characteristic illustrates the capability of the software to perform particular tasks within a specified time frame using minimum resources. This can be measured in terms of response time of the system, processing speed, resource usage, and scalability under different conditions. This characteristic is considered to ensure perfect performance with adequate economy regarding the resources. This saves time, energy, and money.

3. **Compatibility:** This quality proves the software's ability to work together with other software, systems, or surroundings without conflict. It involves such things as interoperability (ability to exchange and use information) and co-existence (ability to perform in a shared environment without deteriorating the performance of other systems) and compliance with standards that support compatibility.

4. **Usability:** Usability is the degree of effectiveness, efficiency, and satisfaction with which specified users can learn to use software, prepare inputs for it, and interpret its outputs. It therefore strongly emphasizes user satisfaction and the achievement, through effectiveness, efficiency, and satisfaction, of specified goals within a particular context of use. Understandability, learnability, operability, attractiveness, and observance of user expectations are included.

5. **Reliability:** It is the degree of performance of software when it is operated under specified conditions for a certain time. This software measure consists of the ability of software to perform its required functions without failure, recover from failures, and other specific aspects. Among these are maturity, or the number of failures that it experiences over a period of time; fault tolerance; recoverability; and availability.

6. **Security:** This quality involves the ability of the software to ensure the protection of information and data from unauthorized access or malicious attacks. This characteristic consists of the following: confidentiality, integrity, accountability, and authenticity.

7. **Maintainability:** This is the degree to which software can be changed to correct defects, meet new requirements, make it more reliable, improve performance or other attributes, or make it easier to understand, diagnose, and fix faults. It includes modularity, reusability, analyzability, modifiability, and testability. These sub-characteristics ensure that the software remains both useful and usable throughout its lifecycle.

8. **Portability:** This sub-characteristic measures how well software performs in environments that are different from its manufacturing environment without requiring undue extra implementation effort. This quality includes adaptability, installability, and replaceability so that the software can be installed in multiple systems without much reconfiguration or redevelopment.

Generally, ISO/IEC 25010:2011 is a solid framework for the evaluation and support of the quality of software products. It lays down criteria very lucidly for good-quality software and facilitates developers, vendors, and users in establishing that the software product is of requisite standards and is ensured to perform the intended functions with due emphasis on functionality, reliability, usability, and performance.

Alternative Frameworks and Standards for Software Evaluation

ISO/IEC 25010:2011 is widely recognized for its comprehensive approach to software product quality evaluation [52]. However, various other frameworks and standards also provide valuable guidelines for assessing different software system aspects. These alternatives can be particularly beneficial for specialized systems like CFRSs or to complement the broad criteria set by ISO/IEC 25010:2011.

ISO/IEC Standards ISO/IEC 25010:2011 is a pivotal standard for software product quality evaluation, but numerous related frameworks and standards also offer valuable guidelines for assessing specific qualities. These are especially beneficial for specialized software like RSs, extending the general framework provided by ISO/IEC 25010:2011. To illustrate, several other key standards exist:

- **ISO/IEC 9126**: This standard identifies six important quality characteristics, including functionality, reliability, usability, efficiency, maintainability, and portability. ISO/IEC 9126 [53] serves as a basis for early-stage software quality assessments, helping developers understand and address potential issues early in the development process.

- **ISO/IEC 14598**: This standard provides a thorough methodology for software product evaluation, covering everything from the planning to the execution stages [54]. It includes establishing evaluation requirements, designing evaluation plans, executing these plans, and reporting on the findings. ISO/IEC 14598 supports structured organizational evaluations to ensure software products meet quality standards and user expectations. These standards collectively facilitate a structured approach to enhancing software quality across various aspects and stages of development.

- **ISO/IEC 25012**: This standard focuses on data quality and defines twelve critical characteristics for evaluating data processed and stored by software systems [55]. These characteristics include accuracy, completeness, consistency, credibility, currency, among others. ISO/IEC 25012 is particularly beneficial for data-driven applications, ensuring the adoption of high-quality standards for data, thus supporting safe and sound decision-making.

- **ISO/IEC 12207**: This standard details the processes essential for managing and executing all stages of the software lifecycle, from development to retirement [56]. It addresses configuration management, quality assurance, and maintenance, emphasizing continuous enhancement throughout the software development process. ISO/IEC 12207 aims to standardize software processes, helping organizations improve the effectiveness of their software development practices.

Other Frameworks and Models Other methods include several frameworks that provide extensive guidelines on how to improve development and capability processes, playing a crucial role in enhancing software quality management. These include:

- **CMMI (Capability Maturity Model Integration)**: CMMI is a widely used framework that helps organizations improve their performance across various dimensions [57]. It offers comprehensive guidelines covering software development, project management, and quality assessment. The model delineates five maturity levels, from initial to optimizing, aiding organizations in developing efficient processes to enhance maturity and consistently produce high-quality products.
- **IEEE 1061**: This standard outlines a comprehensive approach to establishing software quality metrics [58]. It assists organizations in defining, implementing, and effectively using software quality metrics. The methodology details the relevance of metrics to stakeholders' needs, data analysis, and the use of data to make quality-related decisions. This systematic approach helps achieve measurable quality improvements that target the organization's quality objectives directly.
- **SPICE (Software Process Improvement and Capability Determination Model)**: SPICE provides a method for assessing the maturity and capability of software development processes [59]. It defines a structured process for evaluating these processes against a set of improvement and capability determination guidelines. This framework helps organizations understand the strengths and weaknesses of their development processes systematically, aiding in enhancing strengths and thus improving the software developed.

2.1.11 Evaluation Metrics

In this research context, we utilize a comprehensive set of performance measurements including Recall, Normalized Discounted Cumulative Gain (NDCG), Shannon Entropy, Popularity, Gini Index, Log Loss, Root Mean Square Error (RMSE), and Mean Absolute Error (MAE). We briefly describe each measurement as follows:

$$\text{Recall@k} = \frac{rec_i\text{@k}}{rel_i}, \tag{2.6}$$

where CFRS model predicts rec_i@k as the top k number of recommended items, which are true positives. rel_i is the relevance (binary or real number) of the item at index i. The primary purpose of recall@k in assessing the performance of a recommendation system is to retrieve relevant items, highlighting the proportion of relevant items that were recommended in the top k recommendations.

$$\text{NDCG@}k = \sum_{i=1}^{k} \frac{2^{rel_i} - 1}{\log_2(i+1)}, \tag{2.7}$$

where rel_i is 1 if the item at index i is relevant and 0 otherwise. NDCG treats the recommended list of items as a ranked list and penalizes the relevant items with lower rank in the recommended list.

$$\text{ShannonEntropy} = -\sum_{i=1}^{n} p(x_i) \log p(x_i), \tag{2.8}$$

where $p(x_i)$ represents the probability of occurrence of the i-th element in the dataset. This metric measures the uncertainty or randomness in the recommendations.

$$\text{Popularity} = \frac{\sum_{i \in \text{rec}} \text{pop}(i)}{|\text{rec}|}, \tag{2.9}$$

where pop(i) is the popularity score of the i-th recommended item, and $|\text{rec}|$ is the total number of recommendations.

$$\text{Gini Index} = \frac{\sum_{i=1}^{n}(2i - n - 1)p(x_i)}{n}, \tag{2.10}$$

where $p(x_i)$ is the proportion of times the i-th item is recommended, and n is the total number of items. This index measures inequality among values of a frequency distribution (e.g., levels of income).

$$\text{Log Loss} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)], \tag{2.11}$$

where y_i is the actual label of the i-th instance, and $\hat{y}_i$ is the predicted probability. This metric penalizes incorrect classification predictions.

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}}, \tag{2.12}$$

where $\hat{y}_i$ and y_i are the predicted and actual values respectively. RMSE measures the average magnitude of the error.

$$\text{MAE} = \frac{\sum_{i=1}^{n}|\hat{y}_i - y_i|}{n}, \tag{2.13}$$

where $\hat{y}_i$ and y_i are the predicted and actual values respectively. MAE provides the absolute difference between predicted and actual values for all predictions.

2.1.12 Benchmark Datasets

We demonstrate and analyze the performance of the proposed approach on four benchmark datasets: MovieLens-1M [60], Gowalla [61], Amazon Musical Instruments and Amazon Arts, Crafts and Sewing [62, 63].

- **MovieLens-1M (ML-1M)**: This dataset is part of the MovieLens project, which is a movie recommendation service. It includes ratings from users on movies. The data includes user demographics such as age, gender, occupation, and ZIP code, which are useful for personalized movie recommendations. The ratings are on a 5-star scale.

- **Gowalla**: This is a dataset from Gowalla, a location-based social networking site where users share their locations by checking in. The dataset spans from February 2009 to October 2010 and includes check-ins by users. It also features an undirected friendship network, represented with edges.

- **Amazon Musical Instruments (MI)**: This dataset comprises reviews of musical instrument products sold on Amazon. It is useful for sentiment analysis, recommendation systems, and other E-commerce research. The data typically includes product IDs, user IDs, ratings, and text reviews.

- **Amazon Arts, Crafts, and Sewing (ACS)**: Similar to the Musical Instruments dataset, this one focuses on products in the arts, crafts, and sewing categories on Amazon. It includes reviews and ratings, which are useful for analyzing consumer preferences and behaviors in this specific market segment.

- **MovieLens (ML-100k)**: The ML-100K dataset [60] is created by the GroupLens Research Project and includes 100,000 ratings from 943 users on 1,682 movies, along with users' demographic details. These ratings were collected through the MovieLens website from September 1997 to April 1998. It plays a crucial role in improving recommendation systems.

All explicit data are binarized and interpreted as implicit feedback. Dataset statistics are summarized in Table 2.3. We see that the Gowalla and Amazon datasets have the highest sparsity level (i.e. ratio of observed to total interactions).

Table 2.3: Summary statistics of datasets.

Dataset	#Interactions	#Users	#Items
ML-100K	100,000	943	1,682
ML-1M	1,000,209	6,040	3,900
Gowalla	1,027,370	29,858	40,981
MI	1,512,530	120,400	903,330
ACS	2,875,917	303,426	1,579,230

2.1.13 CFRS Baseline Methods

We measure the performance of our proposed solutions alongside several CFRSs, which include the Hybrid Method of Linear and Non-linear Collaborative Filtering (HMLET) [18], Hypergraph Contrastive Collaborative Filtering(HCCF) [64], Simple Graph Contrastive Learning Method (SimGCL) [65], Spectral Collaborative Filtering (SpectralCF) [28], Neural Graph Collaborative Filtering (NGCF) [61], Generalized Matrix Factorization (GMF) [66], Neural Matrix Factorization (NeuMF) [66], and Light Graph Convolutional Network (LightGCN) [23]. We mainly select the comparison methods based on their performance and popularity as Benchmark CFRSs. Here's a brief presentation of these baseline methods:

- **HMLET** [18] is based on Graph Convolutional Networks (GCN) and uses a smart selection module to choose between linear and non-linear steps for spreading the embeddings of each user or item node. This makes the recommendations better by adjusting to the specific features of the nodes.
- **SpectralCF** [28] is an advanced recommendation model that uses collaborative filtering in a user-item graph's frequency domain. It makes use of a special filter defined by the graph's Fourier basis.
- **NGCF** [61] is a framework for collaborative filtering on a graph. It combines user-item interactions in the embedding process by looking at higher-level connection information gathered from multiple layers of embedding spreading using a kind of graph convolutional networks.
- **LightGCN** [23] is a streamlined version of neural graph collaborative filtering based on plain graph convolution. It takes out non-linear activation functions and combines the weight matrices between layers.
- **HCCF** [64] is a self-supervised recommendation framework that addresses the challenges of over-smoothing and sparse supervision signals in collaborative filtering.
- **DGCF** [67] (Deoscillated Graph Collaborative Filtering) is a novel RS model that tackles challenges in CFRS methods. By introducing cross-hop propagation layers and locality-adaptive layers, DGCF breaks the oscillation problem caused by the bipartite structure and varying locality, while adapting information propagation to the density of nodes. The DGCF algorithm is mainly based on LightGCN [23].
- **GMF** [66] uses non-linear functions to generalize MF to a non-linear application. MF learns from user-item interaction patterns and represents both users and items as vectors.

- **NeuMF** [66] is based on both Multilayer Perceptron (MLP) and GMF. In other words, it integrates the linearity of MF with the non-linearity of MLP to define user–item latent factors.
- **DiffRec** [19] utilizes a denoising approach within its generative process to model user interactions, emphasizing a unique method to reconstruct and interpret interaction data effectively.
- **NCL** [20] leverages Neighborhood-enriched Contrastive Learning within graph collaborative filtering to enhance recommendation accuracy, especially focusing on items with few interactions by incorporating neighborhood information.
- **XSimGCL** [21] adopts a noise-based embedding augmentation strategy in its contrastive learning process, aiming to improve the robustness and effectiveness of recommendations even with sparse data.
- **ENMF** [22] proposes an efficient method to learn neural recommendation models using the entirety of training data without relying on sampling, ensuring a thorough learning process.
- **RecVAE** [24] enhances the Mult-VAE approach by integrating a composite prior distribution for latent codes and utilizing a novel training strategy based on alternating updates to optimize performance.
- **SGL** [25] focuses on strengthening node representation learning through self-discrimination techniques, aiming to enhance the recommendation systems by improving the self-similarity of items within the latent space.
- **SimGCL** [26] creates augmented views by modifying the network topology to increase the homophily ratio, thereby enhancing the model's ability to generalize from the given user-item interactions.
- **SimpleX** [27] incorporates a cosine contrastive loss within a unified collaborative filtering model, aiming to refine the distinction and similarity assessment between items and users in the recommendation process.

2.2 Literature Review

The Literature Review section reviews prior research on CFRSs and discusses important challenges and progress in the field. It examines critical problems such as the cold start problem, long tail effects, and diversity challenges, as well as evaluation techniques and reproducibility issues.

It also compares various algorithms, focusing on innovations, evaluation methods, and how effective they are at solving the main problems. Some chapters in this thesis present more details on related work, looking closely at specific aspects. This section gives a broad view of the current research on CFRSs and how solutions are developing over time.

2.2.1 Challenges and Progress in CFRSs

RSs, particularly CFRSs, have greatly influenced the E-commerce sector by personalizing user interactions and enhancing the shopping experience. However, CFRS continues to face challenges, especially with regard to new users and items, as well as accommodating users with unique preferences. This section examines these issues, their impacts, and possible solutions, focusing on the cold start problem, the long tail problem, and the absence of comprehensive frameworks.

The Cold Start Challenge

Zheng et al. [28] proposed Spectral Collaborative Filtering (SpectralCF), a model using spectral graph convolution to figure out high-order connectivity in user-item interaction graphs. This method worked better in warm-start scenarios by using the global structure of interaction data. However, it had problems with the cold-start issue since it needed enough interaction data to build the graph. The computationally heavy eigenvalue decomposition also made it less scalable, and the model didn't use metadata, which might have helped with the cold-start challenges.

Wang et al. [61] introduced Neural Graph Collaborative Filtering (NGCF), which expanded collaborative filtering using graph neural networks (GNNs). NGCF was made to capture higher-order user-item relationships through embedding propagation in interaction graphs. Although it was good for warm-start cases, NGCF didn't do well in cold-start scenarios because it relied too much on past interaction data. Sparse graphs, which are typical in cold-start situations, made the problem worse by increasing overfitting and hurting the model's performance.

He et al. [23] created LightGCN to improve efficiency, which is a simpler graph-based collaborative filtering model. They removed extra operations like transformations and non-linear activations, focusing only on embedding propagation. LightGCN was computationally fast and performed fine in warm-start situations, but it wasn't great for cold-start cases as it only depended on user-item interaction data without using metadata.

Yu et al. [21] worked on XSimGCL, which is a graph contrastive learning framework that aims to make recommendation processes easier. Researchers added unique augmentations to simplify graph contrastive learning and still got good performance. Even so, XSimGCL depended too much on interaction data, which made it not that helpful in extreme cold-start cases where metadata or other information could have been more useful.

Lin et al. [68] introduced TDRO, which aimed at solving temporal feature shifts in cold-start items with distributionally robust optimization. They added factors for worst-case and shifting to capture time trends in item features. This made generalization better for cold-start scenarios, though tuning the robustness parameters carefully was needed to balance computational efficiency with performance.

Huang et al. [69] worked on LLM-InS, a simulator using large language models to create synthetic user-item interactions for cold-start items. By making interactions based on item content, LLM-InS turned cold-start items into warm-start ones, allowing for better recommendations. But, how well it worked depended on how realistic the generated interactions were, which added some extra computational load.

Monteil et al. [70] introduced MARec, a framework that aligned metadata with user-item interactions to tackle the cold-start problem. They improved matrix factorization and autoencoder models by adding metadata, making them better for cold-start scenarios. However, the success of this method relied a lot on having good quality metadata and needed extra steps for preprocessing various data sources.

In summary, methods like SpectralCF, NGCF, and LightGCN perform well in warm-start cases but face difficulties with cold-start problems because they rely heavily on interaction data. Newer approaches such as XSimGCL, TDRO, LLM-InS, and MARec address these challenges by using techniques like contrastive learning, metadata, and synthetic interactions. While these methods show improvements, challenges such as scalability, metadata quality, and computational efficiency still need to be addressed.

Long Tail Challenge

Since 2020, advancements in addressing the long tail problem in RSs have introduced new solutions focused on enhancing diversity and accuracy for long-tail items. These advancements not only address the challenges of the long tail but also work to reduce issues like evaluation overfitting and the need for structured problem-solving methods.

Liu and Zheng [71] developed TailNet, a new network architecture for session-based RSs. TailNet enhances long-tail recommendation performance while maintaining accuracy, with successful tests on real-world

datasets.

Achary and Patra [72] suggested a graph-based hybrid method that improves recommendations for long-tail items, showing better performance compared to existing systems in this area.

Wang et al. [73] proposed CORE, a cosine-pattern-based technique that provides benefits in accuracy, flexibility, and scalability, with strong performance in long-tail recommendations.

Sigova et al. [74] studied data-driven recommendations for long-tail items, presenting solutions within financial RSs. Their approach aims to broaden the customer base by increasing the accuracy of financial recommendations, using long-tail data from network users.

Shafiloo et al. [75] examined users' changing preferences to reduce the negative effects of including long-tail items in recommendation lists. Their method improved both accuracy and diversity, achieving up to 91% accuracy improvement.

Gupta and Katarya [76] tackled the data sparsity issue by creating dynamic fruit recommendations from point-of-sale data, using deep learning methods such as LSTM within a Recurrent Recommender Network (RRN). This approach provides an innovative method to mitigate data sparsity.

Elattar and Fouad [77] offered a comprehensive survey on RS challenges and solutions, summarizing recent achievements and directions aimed at solving the long tail problem.

In the context of financial RSs within social networks, Klioutchnikov and Klioutchnikova [78] have progressed by integrating "long-tail" technology. This approach uses larger databases to serve both general and personalized user requests, helping to address the long tail problem .

Zhao et al. [79] introduced LOT-CRS, a framework designed to alleviate the long-tail problem in conversational RSs. LOT-CRS uses a balanced dataset to improve diversity and enhance recommendations for long-tail items.

Yang et al. [80] introduced LOAM, which improves long-tail session-based recommendations through techniques like Niche Walk Augmentation and Tail Session Mixup, showing significant advancements in managing user interest drift and improving prediction accuracy.

Jangid and Kumar [81] conducted a review of deep learning approaches aimed at addressing cold start and long-tail challenges in recommendation systems. Their work classifies methods based on whether they deal with user-level or item-level long-tail distributions. The study highlights the importance of hybrid models that combine collaborative and content-based filtering to handle sparse interactions. However, it lacks experimental evaluations of newer methods, leaving gaps in understanding their real-world applicability.

Wu et al. [82] presented CoRAL, a framework that merges collaborative filtering with retrieval-augmented large language models to enhance long-tail recommendations. The model uses pre-trained language representations alongside collaborative filtering signals to tackle sparsity issues in user-item interactions for less popular items. Experimental findings indicate notable improvements in prediction accuracy for long-tail items compared to baseline models. However, the framework's reliance on computationally intensive large language models might hinder its scalability for real-time use.

Liu et al. [83] explored the potential of large language models (LLMs) to enhance sequential recommendation tasks, particularly for long-tail users and items. Their proposed framework incorporates sequential patterns and contextual information to better capture preferences for infrequently interacted items. While the approach successfully bridges short-term and long-tail interest modeling, the inclusion of sequential data adds significant complexity, which could pose challenges for implementation in environments with limited computational resources.

These studies contribute to the development of CFRSs capable of handling long-tail items by employing diverse strategies, from leveraging sparse data to accommodating dynamic user preferences. Nonetheless, gaps remain, particularly in addressing overfitting during evaluation and creating structured, scalable methodologies. The dependency on user data or evolving preferences fails to provide a fully consistent solution to the complexities of managing long-tail items.

Diversity Challenge

Efforts to improve diversity in CFRS recommendations have led to various innovative approaches, each addressing this challenge from different perspectives. These strategies include new similarity measures, semantic analysis, group-based preferences, and hybrid models, all aimed at providing users with a wider array of recommendations.

Gazdar and Hidri [84] introduced a novel similarity measure to enhance recommendation diversity. Although their metric effectively captures subtle user preferences, its broader application could be hindered by potential overfitting and the lack of a structured implementation method.

Pujahari and Sisodia [85] focused on the aggregation of preference relations to increase diversity in group recommendations. While this method enhances group satisfaction, it presents challenges in complexity and scalability, which stand out as notable weaknesses.

Fkih [86] conducted a comprehensive comparison of various similarity measures to identify those best suited for diverse CFRS scenarios. The study provides useful insights for improving diversity, but questions around the generalizability of these measures and the need for a structured integration approach remain open for further exploration.

Bobadilla et al. [87] examined deep variational models, introducing stochasticity into CF's latent space to potentially increase diversity. This creative use of deep learning is a notable strength, though the complexity of these models raises concerns about overfitting and underscores the need for a structured deployment approach.

Alhijawi et al. [88] developed hybrid semantic-based CFRSs aimed at enriching recommendations through semantic analysis. While this approach benefits from semantic enrichment, it heavily relies on available semantic information and lacks a holistic methodology applicable across diverse scenarios, marking it as a limitation.

Widayanti [89] merged CF with content-based filtering to utilize item content for enhancing diversity. This hybrid approach offers promise for more varied recommendations but faces challenges in integrating distinct filtering techniques and the risk of overfitting to content features.

Bobadilla et al. [33] studied matrix factorization models and evaluated factors such as accuracy, novelty, and diversity to help select models for different recommendation tasks. Their research mostly focused on traditional matrix factorization and didn't fully consider the issues posed by newer deep learning methods.

Noel [90] used reinforcement learning to increase the diversity of pre-trained models without requiring retraining. This approach is notably adaptable and efficient in promoting diversity. However, it may encounter issues like evaluation overfitting and lacks a universal framework suitable for different CFRS scenarios.

Sharma et al. [91] presented an algorithmic engine aimed at helping users discover diverse content. The approach focused on changing user preferences dynamically and promoting exploration of different domains. However, the method might require significant computational power, which could make it less practical for large systems.

Slokom and Hollink [92] suggested a pre-processing strategy that changes user profiles to offer exposure to more categories and content types. This approach is adaptable to different systems but its results can depend heavily on the type of data and how users interact with the system.

Collectively, these studies add to the discussion on diversity in CFRS recommendations by presenting varied methodologies with potential impact. Yet, they also bring attention to ongoing issues in establishing robust evaluation frameworks and structured methodologies that can scale and support diverse CFRSs. Filling these gaps, especially in covering an extended range of items and creating comprehensive, adaptable strategies, remains a key focus for future research.

Evaluation Overfitting and Reproducibility Challenges

Research in CFRSs has faced recent criticism, particularly regarding study quality, generalization of results, and evaluation overfitting problems. Questions are being raised about whether the field is genuinely progressing

or simply repeating old patterns.

Dacrema et al., in their RecSys 2019 paper titled "Are We Really Making Much Progress? A Worrying Analysis of Recent Neural Recommendation Approaches" [6], examine claims of advancements in neural recommendation models. They argue that when these methods are evaluated under strict and standardized protocols, the improvements over traditional CFRS techniques are minor. Their work stresses the need for consistent benchmarking to fairly measure progress.

In another study, "A Troubling Analysis of Reproducibility and Progress in Recommender Systems Research", published in ACM Transactions on Information Systems [7], the same authors discuss challenges in reproducing results. These difficulties arise from unclear methodology, restricted dataset access, and the lack of shared codebases. They advocate for open science practices, emphasizing transparency and reproducibility as essential for the advancement of RSs' research.

Zhao and Pi [93] address the impacts of rating errors on the evaluation of CFRS methods, proposing a simulated rating experiment to analyze effects on common quality metrics. This study raises concerns about the reliability of current evaluation practices and their ability to genuinely reflect algorithm performance.

In a similar vein, Juan et al. [94] survey collaborative filtering algorithms, tackling issues like data sparsity and cold start problems. Their exploration into a hybrid algorithm combining KNN and XGBoost models for improving personalized recommendations underscores the quest for more robust solutions, albeit questioning their adaptability and scalability.

Kumar et al. [95] present an empirical analysis comparing collaborative and content-based systems, aiming to guide developers in algorithm selection. This analysis spotlights the need for a more nuanced understanding of algorithm applicability based on specific needs, emphasizing evaluation challenges.

Xiao et al. [96] focus on an improved similarity-based algorithm for item rating prediction, highlighting the ongoing efforts to enhance accuracy through methodological advancements. However, the quest for more accurate predictions brings to light the underlying issue of evaluation overfitting, questioning the generalizability of these improved methods.

Zhang et al. [97] provide an overview of CFRS algorithms, pointing out the evaluation issues that plague current research. Their work adds to the growing concern over how CFRSs are assessed, calling for a reevaluation of success metrics beyond traditional accuracy measures.

Wang [98] delves into the data sparsity challenge, offering insights into how this fundamental problem affects the reliability and applicability of CFRSs. Wang's work echoes the broader community's concern about the scalability and real-world application of CFRS models.

Ghazanfar and Prügel-Bennett [99] propose a hybrid model that attempts to mitigate some of CFRS's inherent limitations. While innovative, the practical implementation of such models and their performance in varying contexts remain underexplored.

Hu [100] explores multicriteria approaches to broaden CFRSs' scope. This shift towards considering various user preferences and criteria underscores the complexity of truly capturing user needs and preferences.

Recent critiques further extend to the integration of trust mechanisms and temporal dynamics in CFRSs, as explored by Lei Zhang et al. [97]. Their proposed UR system highlights efforts to refine CFRS recommendations but also raises questions about these methods' adaptability and long-term effectiveness.

Kuanr and Mohapatra [101] survey assessment methods for evaluating RSs, organizing various metrics and exploring their relationships. This work underscores the need for comprehensive approaches to assess the multifaceted impacts of CFRSs.

Anelli et al. [102] analyzed graph-based collaborative filtering models, highlighting issues with reliance on unverified baseline results and the lack of standardized evaluation frameworks. These gaps limit comparisons across studies and result in poor generalization to real-world applications, particularly when datasets are dynamic.

Together, these studies represent a critical discourse on the state of CFRSs, highlighting significant issues

such as evaluation overfitting, the need for greater methodological rigor, and the quest for more generalized and scalable solutions. Addressing these concerns is pivotal for the advancement of CFRSs, ensuring they not only perform optimally in controlled environments but also translate effectively into real-world scenarios.

Limitations of Current Frameworks for Enhancing CFRSs

The field of RSs is seeing growth in innovative frameworks designed to improve machine learning and AI applications. These systems address challenges like accuracy, privacy, user satisfaction, scalability, and resistance to manipulation.

Wenqiang Lei et al. [103] study conversational RSs that use natural language processing for personalized recommendations. Despite their promise, challenges remain in integrating traditional engineering practices and comprehensive evaluation.

Himan Abdollahpouri et al. [104] introduce a multi-stakeholder framework aimed at balancing the needs of consumers, providers, and platforms. While it focuses on fairness and diversity, questions about scalability and applicability in different domains remain open for further exploration.

Huang et al. [105] investigate self-supervised learning to improve recommendation accuracy and efficiency. By leveraging unlabeled data, this approach shows potential but struggles to directly handle privacy issues and user satisfaction, leaving gaps for further research.

Insights from studies like Idakwo et al. [106], Subhash [107], and Piyadigama and Poravi [108] show progress in the field. However, gaps still exist in privacy integration, scalability, and empirical validation across industries.

Structured problem-solving strategies suggested in these works are not fully utilized, leaving room to improve system design. Furthermore, current research does not align well with quality assessment frameworks like ISO/IEC 25010:2011, indicating the need for standardized evaluation practices.

In domains like E-commerce, as noted byZheng and Wang [109], empirical validation is limited, pointing to the need for broader testing. Meanwhile, Lin et al. [110] explore reinforcement learning to enhance RSs, suggesting potential for new innovations.

Recent research into CFRSs focuses more on critical analysis of structural and methodological issues. Studies by Kong et al. [18], Wang et al. [19], and Lin et al. [20] discuss evaluation overfitting, generalization, and methodologies. While progress is evident, challenges related to quality, scalability, and evaluation frameworks persist, leaving space for further improvement.

Other researchers have noted difficulties in replicating published results, pointing to issues such as incomplete descriptions of methodologies, limited access to datasets, and the lack of shared codebases [6, 7]. Reproducibility is considered crucial for scientific progress, and open science practices have been suggested as a solution to improve research transparency and validation.

One study explored the effect of rating errors on evaluating CFRSs methods [93]. By simulating rating errors, the study measured their impact on quality metrics and concluded that existing methods cannot adequately evaluate CFRS algorithms under real-world conditions.

Another paper addressed the challenges of data sparsity and cold start in CFRSs [94]. The researchers proposed a hybrid model combining KNN and XGBoost to improve personalized recommendations. However, they questioned the adaptability and scalability of the approach.

Xiao et al. [96] studied improved similarity-based algorithms for predicting item ratings, aiming to enhance accuracy through methodological improvements. However, their reliance on refined evaluation methods raises concerns about overfitting and the generalizability of their findings.

Zhang et al. [111] reviewed CFRS algorithms and identified significant challenges in evaluation processes. Their study pointed out flaws in current success metrics and emphasized the importance of revising evaluation measures beyond traditional accuracy-focused approaches.

Wang [98] focused on data sparsity, a key issue affecting CFRSs' reliability and scalability in practical use.

This challenge remains widely recognized as a major limitation in the field.

Ghazanfar and Prügel-Bennett [99] introduced a hybrid model to address the inherent limitations of CFRSs. While the model shows promise, it is still in the early stages of development and has yet to demonstrate practical effectiveness across diverse scenarios.

Hu [100] examined multi-criteria approaches for extending the functionality of CFRSs but acknowledged difficulties in effectively addressing diverse user preferences.

Zhang et al. [112] proposed enhancements for CFRS recommendations but raised concerns about the adaptability and long-term viability of advanced techniques.

Kuanr and Mohapatra [101] cataloged evaluation metrics for RSs, emphasizing the importance of comprehensive frameworks to capture the diverse impacts of CFRSs.

Kumar et al. [95] conducted an empirical study comparing collaborative and content-based systems to help developers choose appropriate algorithms for specific challenges. Their findings highlighted the need for more detailed insights into algorithm applicability and evaluation issues.

Ren et al. [113] presented SSLRec, a self-supervised learning framework aimed at handling the problems of sparse and noisy data in RSs. Although SSLRec offers a structured and adaptable platform for testing SSL-based recommenders, its usefulness in a wide range of applications is yet to be established.

Deng et al. [114] introduced JNCF+R, a neural collaborative filtering model that integrates rating reliability to boost recommendation accuracy and robustness. The model applies fuzzy set-based noise detection and K-means clustering to achieve balanced predictions. While it improves Recall and NDCG, it depends on manual threshold tuning and performs less effectively in cases with only implicit feedback.

In conclusion, these studies collectively underscore several critical issues in CFRS research. Key concerns include overfitting in evaluations, inadequate representation in recommendations, and the absence of structured development frameworks. Resolving these problems is essential for advancing CFRS' efficiency in real-world scenarios rather than merely controlled experimental environments.

Evaluation Challenges in CFRSs

The literature on evaluation methodologies for RSs has seen significant contributions spanning various aspects such as current evaluation practices, their limitations, and proposals for enhancing the quality, reliability, and scalability of these evaluations. Notably, ISO/IEC 25010:2011 plays a crucial role in establishing standards for evaluating software product quality, including RSs, which is essential for advancing the field towards more rigorous and standardized evaluation frameworks.

The literature also identifies gaps in current evaluation methodologies, such as the need for methods that are universal, easily implementable, and resource-efficient [115].

Kashamova [116] applied the ISO/IEC 25010 quality model to minimize subjective factors in evaluating software product quality and efficiency, demonstrating the model's applicability across different types of information systems.

Nuzula and Rochimah [117] employed a descriptive quantitative approach to assess the quality of Human Resource Information Systems based on ISO/IEC 25010, focusing on functional suitability and usability. In a different vein, Al Jurdi et al. [118] introduced a neighborhood-based evaluation methodology that aims to provide a deeper understanding of performance variations in RSs beyond what traditional methods offer. This highlights the ongoing efforts to address the limitations of current evaluation practices by exploring new methodologies that can offer more nuanced insights into system performance.

Offline evaluation methods are an important way to assess RSs using pre-existing datasets. Metrics such as precision, recall, Mean Absolute Error (MAE), and Root Mean Square Error (RMSE) are often used to measure performance without requiring real-time interaction. Jadon and Patil [119] discussed the need to include context-aware metrics in offline evaluation frameworks. This helps to simulate user behavior and preferences better while improving the scalability of these systems, even though static datasets have their

limitations.

User-centric evaluations have become more popular as they focus on user satisfaction, trust, and ethical issues. Methods like surveys, interviews, and controlled experiments provide qualitative insights that add to the quantitative metrics. Ekstrand et al. [120] proposed a distributionally-informed evaluation method. This method looks at how utility is distributed among different stakeholder groups, ensuring fairness and inclusiveness and addressing biases in traditional evaluations.

These examples show how scholars are trying to improve how RSs are evaluated. By overcoming current challenges and using standards such as ISO/IEC 25010:2011, the field is moving towards methods that are more reliable, scalable, and focused on quality.

2.2.2 Comparative Review of CFRS Algorithms in Literature

Table 2.4 compares various CFRS algorithms in terms of their statistical significance, techniques, metrics, and their ability to address cold start and long tail problems. The findings are discussed in the following sections.

Statistical Significance and Evaluation Techniques

Some algorithms apply statistical tests like the T-test to validate their results. For example, HMLET [18], NCL [20], ENMF [22], and SGL [25] use this method to show their improvements. However, many algorithms, such as DiffRec [19], XSimGCL [21], and RecVAE [24], do not mention using any statistical tests, which can raise doubts about the reliability of their results.

Techniques and Innovations

The algorithms take different approaches to improve recommendations:

- **Graph-based Approaches:** NCL [20], LightGCN [23], and SpectralCF [28] use graphs to represent user-item relationships. NCL and SGL focus on contrastive learning within graphs, while LightGCN uses simpler propagation techniques.
- **Noise-Augmented Embeddings:** XSimGCL [21] introduces noise-based embedding augmentation for contrastive learning.
- **Generative Techniques:** DiffRec [19] and RecVAE [24] use generative processes to model user interactions more effectively.

Metrics Used for Evaluation

Most algorithms are evaluated using Recall and NDCG, which are common metrics for ranking tasks. Some methods, however, use other metrics:

- **F1-Micro:** SimGCL [26] evaluates performance using F1-Micro, which is better suited for classification tasks.
- **Mean Average Precision (MAP):** SpectralCF [28] includes MAP to provide a broader evaluation of ranking quality.

Addressing Cold Start and Long Tail Problems

Cold start remains a challenge for recommendation systems. Only a few methods, like XSimGCL [21] and SpectralCF [28], tackle this issue. Similarly, some algorithms, such as NCL [20], XSimGCL, and SGL [25], aim to handle the long tail problem, where data distribution is uneven or sparse.

Insights and Limitations

The analysis highlights some important gaps and strengths in the algorithms:

- **Limited Cold Start Handling:** Few algorithms address the cold start problem, showing an area where more research is needed.
- **Overuse of Recall and NDCG:** While these metrics are useful, relying too much on them might not reflect the diversity of recommended items.
- **Inconsistent Use of Statistical Testing:** Many methods do not apply statistical significance tests, making their results harder to compare.

This review shows the variety of techniques used in modern CFRS methods and their limitations. Future work could focus on better validation techniques, new evaluation metrics, and stronger methods for cold start and long tail problems.

Table 2.4: Comparative Analysis of CFRSs' Algorithms

Algorithm	Statistical Significance	Technique Description	Metrics Used	Cold Start Handling	Long Tail Handling
HMLET [18]	T-test	Direct optimization of alignment and uniformity in user and item embeddings	Recall, NDCG	No	No
DiffRec [19]	Not Used	DiffRec model using a denoising approach for the generative process of user interactions	Recall, NDCG	No	No
NCL [20]	T-test	Neighborhood-enriched Contrastive Learning (NCL) approach in graph collaborative filtering	Recall, NDCG	No	Yes
XSimGCL [21]	Not Used	XSimGCL method employs a simple noise-based embedding augmentation for contrastive learning	Recall, NDCG	Yes	Yes
ENMF [22]	T-test	Learn neural recommendation models from the whole training data without sampling.	Recall, NDCG	No	No
LightGCN [23]	Not Used	LightGCN utilizes linear propagation on the interaction graph to derive user and item embeddings.	Recall, NDCG	No	No
RecVAE [24]	Not Used	Improve Mult-VAE through composite prior distribution for the latent codes, and a new approach to training based on alternating updates.	Hit Ratio, NDCG	No	No
SGL [25]	T-test	Reinforce node representation learning via self-discrimination	Recall, NDCG	No	Yes
SimGCL [26]	Not Used	Generate augmented views with a higher homophily ratio at the topology level by adding or removing edges.	F1-Micro	No	No
SimpleX [27]	Not Used	Incorporate cosine contrastive loss (CCL) into a simple unified CFRS model.	Recall, NDCG	No	No
SpectralCF [28]	Not Used	Learn from the spectral space of Bipartite Graphs	Recall, MAP	Yes	No

Chapter 3

A Methodological Framework for Enhancing CFRSs

3.1 Introduction

Looking back to section 1.6, acknowledging and mitigating these challenges in a single CFRS requires structured planning, beyond rule-of-thumb practices. Indeed, our multidisciplinary approach in this research is based on an analogy with systems engineering and software engineering, reflecting the lack of tailored frameworks for developing CFRSs. Accordingly, this chapter presents a multidisciplinary methodological framework for enhancing CFRSs. This begins by the introduction of the relevance and the elements of the methodological frameworks in the context of CFRSs. The discussion then proceeds to specify how system block diagrams and functional analysis are actually able to translate into the enhancement of system design and development. It then introduces the application of TRIZ (Theory for the Solution of Inventing Problems) to address the contradictions in CFRS models. It brings up literature review and related work, a comparative analysis of all those methodologies, and practical use. Further, the work elaborates on the processes of data preprocessing, model training evaluation, and statistical analysis, where at the end, the effectiveness of the strategies of CFRS optimization is compared.

3.1.1 Introduction to Methodological Frameworks

As explained in Section 1.6, a methodological framework is a structured scientific approach that offers a coherent and systematic guide to addressing complicated issues in the scientific world. According to Omitaomu et al. [121], this framework refers to an ordered constellation of concepts, methods, and practices that steer a project right from the start to implementation and final evaluation. Thus, it organizes how available knowledge is captured, treated, adopted, and prepared for future utilization. The significance of the framework in a CFRS is even more vital due to the system's multifaceted nature [122]. Since CFRSs entail the manipulation of huge volumes of data, various events by users, and robust technology, adapting a well-defined approach ensures that each segment is optimized and that the components are integrated effectively.

3.1.2 Key Components of a Methodological Framework

The framework typically integrates several key components [123]:

- **Theoretical Models:** These provide the foundational concepts and theories that explain the system's design and functionality.
- **Design Principles:** These principles guide the system architecture and the interaction of different components, ensuring that the system is robust and scalable.

- **Methodologies:** Detailed methodologies for data collection, analysis, and processing that guarantee data integrity and usability at the same time.
- **Best Practices:** Industry standards and best practices that guide the development process, ensuring that the system is effective.
- **Evaluation Metrics:** Guidelines and metrics for evaluating system performance, making sure that the system meets the required standards.

By incorporating these components, a methodological framework helps researchers and developers in the field of RSs to navigate the complex landscape of technology development, providing a clear path to creating systems that are not only technologically advanced but also user-centric and ethically sound.

3.1.3 The Significance of a Methodological Framework in Developing CFRSs

Having considered the extensive challenges identified in the problem statement[1] regarding CFRSs. Indeed, it is fundamentally necessary to elaborate a comprehensive methodological framework to advance the CFRSs' effectiveness, efficiency, and flexibility. The framework's importance is guided by not only addressing the optimization of traditional metrics such as recall and NDCG but also solving intrinsic issues in CFRSs systems that relate to diversity, fairness, and operational incidences. This section elucidates the various vital functions the methodological framework performs in CFRSs development by illustrating how it addresses each challenge described above to make a well-detailed performance improvement.

- **Generalization, Assimilation, and Dissemination:** This aspect of the framework [124] directly tackles the evaluation and optimization overfitting challenge. The generalization aspect ensures that the solution will perform not only in a controlled and familiar environment but will also overcome overfitting by ensuring performance in real-world uncontrolled environments. The assimilation aspect ensures that the framework integrates new technologies and methodologies to stay up-to-date, which helps address challenges that may arise from user demographic changes or responding to privacy concerns. The dissemination aspect ensures that everyone, from developers to end-users, is supported to understand and work with the recommendation systems.
- **Systematic Development Process:** This process allows developers to follow a development trajectory from data handling to the deployment of the system. The systematic development process reduces the time used in developing solutions while ensuring that the systems are flexible enough to meet diverse user needs and are capable of identifying both mainstream and long-tail preferences.
- **Enhanced Problem Solving:** This framework component is expressly intended to help solve challenges related to diversity, novelty, and systematic fairness in recommendations. Through a systematic approach to identifying, analyzing, and solving issues, this section is suited to strike the right balance between accuracy and recommendation diversity. It highlights the system's capability to recommend relevant and varied content, thereby increasing user satisfaction and engagement. Moreover, the system addresses scalability issues effectively, ensuring that the CFRS can handle large-scale data volumes, which are key to maintaining efficiency and fairness as the system grows.
- **Detection and Resolution of Contradictions and Inconsistencies:** This framework is designed to highlight and resolve contradictions and inconsistencies within the optimization pipeline [125]. For instance, it can demonstrate how efforts to increase processing speed might compromise the quality of recommendations, or how efforts to enhance recommendation diversity might impact accuracy. Through

[1]Further information is available in Section 1.4

the provision of tools and methodologies for resolving these conflicts, the framework ensures holistic optimization of the system.

- **Quality Assurance and Evaluation:** This part of the framework not only enables the application of rigorous standards for performance evaluation and quality assurance but also paves the way for addressing the reproducibility challenges common to most RSs. It ensures the recommendations not only produce consistent and reliable results across different testing environments and data sets but also remain stable over time despite transient variations in data or setup. By comparing the system's performance against well-established metrics, this part of the framework assures current and future performance standards are met. The optimization progressively addresses latent issues like sparsity and overfitting, ensuring the system performs well and delivers high-quality recommendations.

- **Ethical and Fair Recommendations:** The incorporation of ethical guidelines and fairness procedures forms an important part of this framework, addressing fairness and bias challenges [4]. Such protocols make it easier to achieve equitable representation in recommendation lists by ensuring diverse user interests are favored without necessarily depending on popular items. This part of the framework not only builds user trust and satisfaction but also ensures that ethical delivery practices are observed through systematic governance.

In conclusion, the methodological framework forms the backbone of strategic development for CFRSs. It should not be regarded as the tool but a basic part of it, ensuring that the developed system is robust, adaptive, and able to give top-notch user experiences, all the while assuring that technical contradictions and business challenges will be handled effectively.

3.1.4 Enhancing CFRS Design with Systems Block Diagrams

Systems block diagrams are essential tools in systems engineering, offering a clear and structured visual representation of the interconnections and relationships between various components of complex systems [126]. For CFRSs, these diagrams serve as a critical tool for refining system architecture, improving component interaction, and resolving inherent contradictions and problems.

Strategic Role of Systems Block Diagrams in System Design

The strategic implementation of systems block diagrams in CFRSs provides several key benefits:

- **Macro View of System Architecture:** When used with large-scale systems, block diagrams provide a macro view of the RS with insights into how data input, processing modules, and output generation are linked. This view is crucial in ensuring that the system's design is comprehensive, coherent, and logically structured.

- **Improving Design and Component Interaction:** By outlining the major subsystems and their interactions, the diagram can be used to identify areas where the component interaction can improve. This ensures that the architecture is not just optimized for performance but is also designed for scalability and maintainability.

- **Identifying and Resolving Contradictions:** The block diagrams help identify contradictions within the system. An example of such a contradiction could be the need for both speed and accuracy, and another one could be the need for a complex system while maintaining simplicity. Discovering the contradiction during the early stages of development means that the developers can make specific adjustments to the system to eliminate the contradictions.

- **Facilitating System Reproduction and Consistency:** The block diagrams depict the system in a clear, organized manner. This makes it easier to reproduce the diagram for scaling purposes or for use in other environments. The diagram ensures that each reproduction of the system adheres to a standard that addresses the architectural standards and operational requirements. Therefore, the block diagram ensures consistency across different deployments.

- **Streamlining Troubleshooting and Upgrades:** The layout of the block diagram and the integration of components ensure easy identification of potential problem areas. The developers can also conduct updates without affecting other components within the system.

Implementation in CFRSs

Integrating the use of systems block diagrams in designing CFRSs is a directional process aimed at enhancing their structure and performance. The approach includes several key steps:

- **Initial Design Mapping:** Start by mapping out the constituent components of the CFRS, including user profile inputs, algorithmic processing units, and recommendation output modules. This step establishes how each part of the system will be structured.

- **Detailed Interaction Analysis:** Analyze the flow of data between parts to identify any bottlenecks or operational inefficiencies. This analysis helps in pinpointing areas that can be optimized for data processing and algorithm application.

- **Optimization and Simplification:** Use the block diagram to simplify processes by rearranging and eliminating some subsystem processes. This step also focuses on eliminating unnecessary complexities that could hinder system performance or make it volatile.

- **Continuous Refinement:** Regularly revise and update the block diagram to incorporate new technologies and methodologies. This ongoing refinement ensures the CFRS adapts to new challenges and maintains high performance and user engagement.

In essence, systems block diagrams serve as foundational design tools that significantly improve the structure and efficacy of CFRSs. They provide a clear blueprint that ensures the system is robust, adaptable, and consistently delivers high-quality recommendations.

3.1.5 Optimizing CFRSs with Functional Analysis

Functional analysis is used to help improve the utilization and performance of CFRSs. Knowing the work of every tiny component is critical to improving the algorithms and Python code used to run these systems. This microscopic method outlines the actions of individual components [127], contrasting with the macro perspective offered by block diagrams, which show the architecture of the whole system and the connections between components.

Strategic Role of Functional Analysis in System Development

Functional analysis offers several substantial benefits when applied to the development and optimization of CFRSs, particularly in modifying and improving Python code and algorithms:

- **In-depth Component Functionality Assessment:** Functional Analysis starts with a deep analysis into what each component is supposed to do. This helps identify areas where developers might be ineffective or where there are redundancies in the code base. Understanding whether modifications are necessary to achieve efficient functionality or enhanced performance is crucial, and functional analysis aids in this process.

- **Algorithm Optimization:** Functional analysis also aids in understanding how different algorithms fulfill their roles within a system that may be inefficient or overextended. The essence of this understanding is to come up with refined variants or find replacements to ensure both functionality and efficiency.
- **Enhancing Code Efficiency:** Another benefit of functional analysis is streamlining Python code by reducing complexity and eliminating unnecessary functions. This results in faster execution and easier maintenance and debugging of the code.
- **Support for Scalable and Maintainable Code:** Understanding the functions each part of code performs supports more scalable and maintainable coding practices. Developers can modularize the code, making it easier to upgrade and adapt as system requirements evolve.
- **Facilitating Targeted Improvements:** Functional analysis enables developers to make more targeted improvements to the Python code, minimizing adjustments to other parts of the system and ensuring effectiveness and efficiency.

Implementation Strategy for CFRSs

To effectively apply functional analysis in CFRSs, developers should follow these steps:

- **Function Mapping:** Begin by documenting the specific functions of all Python code segments and algorithms. This step involves detailing how data is handled, processed, and utilized to generate recommendations.
- **Code and Algorithm Evaluation:** Evaluate the performance and efficiency of existing code and algorithms against their intended functions. This process helps identify any discrepancies or areas that need enhancement.
- **Code Optimization:** Based on the findings from the functional analysis, refine or rewrite Python code and algorithms. This step focuses on enhancing efficiency, reducing complexity, and improving the overall performance of the system.
- **Continuous Code Refinement:** Regularly revisit the functional analysis to integrate new data, user feedback, and technology trends. Use these insights to make iterative improvements to the code, ensuring that the system remains adaptive and efficient.

In conclusion, functional analysis is indispensable for systematically enhancing the Python code and algorithms in CFRSs. It ensures that every line of code and each algorithm is not only functional but also optimized for performance, maintainability, and scalability, aligning with the dynamic needs of users and technological advancements.

3.1.6 Introducing The Theory of Inventive Problem Solving (TRIZ)

The Theory of Inventive Problem Solving (TRIZ) is an integrated methodology designed to support a systematic approach to innovation development [128]. TRIZ is based on the analysis of an extensive dataset of patented inventions. It suggests that the development of technological systems adheres to unchanging patterns and states that inventive problems can be systematically resolved by using principles implemented in analogous solutions developed in other fields.

Core Framework and Tools TRIZ encompasses a set of tools and principles designed to identify and solve contradictions, the root of inventive problems, without compromises. [129] Key components of the TRIZ framework include:

- **40 Inventive Principles**: These are foundational strategies for eliminating contradictions and stimulating creativity.
- **Contradiction Matrix**: This tool helps target the most relevant principles based on the nature of the contradiction.
- **Patterns of Evolution**: These trends forecast the future development of a system, enabling proactive innovation.
- **Function Analysis**: A method to explore the interactions between the components of a system and find ways to improve its effectiveness.
- **Ideality**: This core concept posits that the best solutions solve problems without creating new ones, optimizing the system to be lighter, cheaper, smaller, or faster.

Principles of TRIZ Among the 40 inventive principles, several are particularly relevant to technological innovation, including *Segmentation*, *Taking Out*, *Local Quality*, and *Dynamicity*. These principles offer various approaches to deconstructing problems, rethinking system components, and identifying innovative solutions.

3.1.7 Applying TRIZ to Resolve Contradictions in CFRS Algorithms

TRIZ can be applied to CFRSs algorithms to improve recommendation systems. It provides a way to find and resolve contradictions that are difficult to handle with regular methods.

A common problem in CFRS algorithms is the trade-off between accuracy and computational requirements. When accuracy improves, the system often becomes slower and harder to scale. This creates challenges for making the system both efficient and effective.

TRIZ principles can address this issue. For example, the Segmentation Principle proposes splitting the algorithm into smaller, simpler parts. These parts can run only when needed or work at the same time to reduce the overall load. The Local Quality Principle suggests improving specific sections of the algorithm for certain users or items. This approach avoids adding too much complexity while boosting efficiency. The Dynamicity Principle also helps by making the algorithm more flexible to adjust based on changing needs.

Using TRIZ in this way makes CFRS algorithms more innovative and adaptable. It helps solve critical challenges and creates better solutions for users.

3.1.8 Tailoring TRIZ for CFRS Algorithms

The direct application of TRIZ principles for CFRSs must result in not only a one-to-one implementation but also a certain adaptation according to the peculiarities of CFRSs. It means that initially one must look into typical issues with CFRS algorithms that stem from the need to balance computation-intensive accuracy, tackle the cold start problem, and address the diversity of recommendations needed to counteract the long-tail effect.

Customization of TRIZ Principles To effectively utilize TRIZ in the context of CFRSs, it's crucial to customize its principles and tools. This customization might involve:

- **Adapting the Contradiction Matrix**: The traditional TRIZ contradiction matrix can be adapted to include contradictions unique to CFRS such as the one between recommendation diversity and personalization. This can help to indicate which TRIZ principles are most relevant to CFRS-related challenges.

- **Specialized Inventive Principles**: All 40 inventive principles are theoretically applicable, but some are likely to be more applicable within the realm of CFRS than others. For example, Principle 15 – Dynamics suggests that an object's physical state should be changed to improve function. This could inspire the dynamic approach concept, where the algorithm's complexity is adapted continuously in real-time based on the current level of demand.

- **Patterns of Evolution**: TRIZ also identifies a number of patterns or models which typify how systems evolve. These can be applied to CFRSs to suggest how the algorithm could develop in the future. Increasing personalization of the target domain or the level of automation applied to adaptation in accordance with user feedback may be predicted.

Innovation in Algorithm Design The goal of tailoring TRIZ for CFRSs is to foster innovation in algorithm design, enabling systems to better meet user needs while efficiently handling the vast and dynamic data inherent in recommendation scenarios. This involves:

- **Systematic Problem-Solving**: The consistent application of the TRIZ toolkit in problem identification and solution within CFRS algorithms should promote a more structured trend in the innovation process, eliminating the use of trial-and-error methods.

- **Overcoming Design Contradictions**: The concept of contradiction in TRIZ will help to focus on solving contradictions. In other words, the better systems become, the more difficult it is to find a balance that will enable dynamic equilibrium and continuous operation. In this way, the effort will be focused on creating more complex, yet user-friendly CFRSs.

- **Enhancing User Experience**: Ultimately, the implementation of TRIZ principles is expected to benefit the user experience by enabling more precise and varied recommendations to be given faster, thereby improving user engagement.

To conclude, customizing TRIZ for the domain of CFRSs allows not only to increase the efficiency of this problem-solving approach but also to unlock creative and innovative ways of algorithmic development. Hence, with the intelligent use of altered TRIZ tools, it becomes possible to overcome the limitations of the CFRSs domain and generate highly advanced CFRSs, particularly with the support of tailored Decision Support Systems (DSSs).

Leveraging Functional Analysis and System Block Diagrams in CFRSs Functional analysis and system block diagrams are essential tools of the TRIZ approach that are particularly insightful when applied to CFRS algorithm development. Using these tools can help engineers understand the system's functioning more thoroughly, and the suitable arrangements of its elements will allow for meaningful improvements.

Functional Analysis in CFRS In the context of CFRS, it is possible to discern the following arrangements: the nature of user input, the functioning of the algorithm itself, and the way it processes the data into recommendations. Based on such analysis, we can observe the imbalances, excesses, and gaps within the system, such as:

- Inadequate personalization capabilities, as the system poorly captures the user's preferences.

- Excessive complexity, including parts of the algorithm that overburden the system with unnecessary computational work.

- Insufficient closing of the loops, with no feedback mechanism to support the recommendations.

System Block Diagrams for CFRS System block diagrams serve as a complement to functional analysis by providing a visual guide to the structure and flow of data within the CFRS. They enable one to see the entire pipeline of operations that occur from collecting initial data to feeding recommendations back to the end-user. When considering CF algorithms, system block diagrams offer insights into the various possible pathways through which data is processed, where potential bottlenecks occur, and where optimization through parallel processing or modular methods can be applied. For developers, this means that the following strategies can be considered:

- Recommendations can be segmented to allow user similarity indices to be computed independently.
- Feedback can be incorporated with minimal effort into the data flow such that the CFRS is adjusted rapidly if users interact with the recommendations.
- Part of the system can be isolated for specialized processing. For example, it's easier to treat a new user or item as a special case rather than as part of the general-case recommendation process.

Enhancing CF with TRIZ Tools Functional analysis and system block diagrams within TRIZ provide an organized approach to the dissection of CF algorithms and provide a better understanding. Moreover, this approach not only points out the weak points but also suggests high-performing solutions in line with ideality and innovation. Consequently, if CFRSs developers utilize the tool in the right way, they can expect an enormous surge in algorithm quality and performance, as well as scaling and customer satisfaction. Accordingly, the use of TRIZ tools, including functional analysis and system block diagrams in its development, assists in the all-inclusive, detailed consideration of the functional dynamics in the system. As a result, this allows for handling contradictions and inefficiency, leading to the development of highly functional and human-centered recommendation systems.

3.2 Related Work

This section reviews optimal RSs in recent proposals. The sources propose developments geared toward enhancing recommendation accuracy, user satisfaction, and responsiveness, taking privacy and trust into consideration. Recent key studies are on self-supervised learning, multi-stakeholder frameworks, conversational RSs, and TRIZ methodologies narrowing the gap between traditional and CFRSs. The research would help in the review of diverse strategies to enhance CFRSs.

Frameworks for Optimizing Recommender Solutions

In the recent landscape of RSs, a series of frameworks proposed in the past several years suggest novel ideas for improving various processes in machine learning and AI. Along with working toward boosting the accuracy of recommendations, those frameworks target privacy issues, user satisfaction, trust and safeguarded actions against manipulators, as well as system stability and responsiveness.

Chao Huang et al. [105] proposed a recommendation system-based framework for self-supervised learning, which could boost recommendation effectiveness in cases when unlabeled data is available. Another multi-stakeholder framework proposed by Himan Abdollahpouri et al. [104] focuses on diversity and fairness of recommendations based on the variety of stakeholder preferences and needs. However, as noted by the authors, central questions about the enhanced scalability and universality of these findings still need to be answered in light of future studies.

Conversational RSs are a trend in RS research and application that leverage natural language processing to engage users in the recommendation process. Lei et al. [103] suggest that this architecture presents challenges in overall evaluation and incorporating traditional engineering approaches and foresee further development in these directions.

Empirical testing, including studies on role-based recommendations by Zheng and Wang [109], is prevalent but not wholly universal, which creates room for further exploration and testing. Lastly, Y. Lin et al. [110] introduce new ways to make reinforcement learning enhance the performance of RSs.

Recent research focused on improving AI's problem-solving capabilities by using TRIZ methodologies. TRIZ-based improvements for AI were explored [42, 43], and the contradictions between functionality and usability were handled with TRIZ [125]. A gamification innovative learning method to TRIZ education was presented [130]. These studies collectively contribute to AI's innovative problem-solving. However, applying these methodologies in diverse realistic scenarios is challenging.

These frameworks, together with the insights from Idakwo et al. [106] on the strengths and weaknesses of collaborative and content-based filtering, Subhash's review [107] on the challenges of RSs, and the emphasis on evaluation metrics highlighted by Piyadigama and Poravi [108], collectively point to a demanding research area. Each framework contributes to highlighting different quality attributes—accuracy, diversity, privacy, and user satisfaction, emphasizing the need to acknowledge these quality attributes. Meanwhile, litrature shows there is a lack of a comprehensive and tailored framework for optimizing CFRSs.

3.2.1 Block Diagrams

Recent studies have emphasized the importance of a system block diagram in the development of intelligent and machine learning applications in various fields. A system block diagram is a core tool because it visually represents the structure and connection of neural networks and simplify their understanding, analysis, and optimization processes.

Doroshenko et al. [36] discussed the application of neuro-evolution tools in the automation of technical control systems, pointing to the simplicity of understanding complex neural network configurations using system block diagrams. Moreover, Benton [37] uses system block diagrams to explain the architecture of a system and the role of each component in a machine-learning system and intelligent application. This simplified understanding is crucial considering that machine-learning and intelligent systems have complex architectures with many components and, therefore, can be processed intuitively.

Additionally, Kuwajima et al. [131] discussed the engineering process of safety-critical machine-learning systems, using a system block diagram to decompose the data-driven process of training the model into requirements, design, and verification processes. This step is critical in various applications such as automated driving where safety considerations should be highly reliable.

Raghavendran and Elragal [38] discussed how to import system block diagrams into low-code machine learning platforms, which allow for the acceleration of automatic machine learning models' development. The ability to rapidly develop and apply such models is crucial for organizations that have to undergo a digital transformation but do not have enough time to accomplish it.

Additionally, Han et al. [132] presented an automated machine learning platform that uses HTN-based plans in a system block diagram shape to produce machine learning pipelines that are appropriate for various tasks. The described feature raises the possibility of applying intelligent or machine learning models by non-experts in these fields, which makes sophisticated data analysis methods more popular.

These studies collectively stress the importance of a system block diagram in developing an in-depth understanding, efficient design, and practical implementation of machine learning and intelligent systems, contributing to major breakthroughs in artificial intelligence.

3.2.2 Functional Analysis

In the domain of machine learning and artificial intelligence, the application and development of functional analysis diagrams have been advanced and promising in the previous years. This tool helps understand complex algorithms and systems in-depth and promotes their increased effectiveness by providing a detailed framework to analyze their separate functions and mutual influences.

The research by Zhang [39] is focused on exploring the enhanced abilities of machine learning algorithms via functional analysis diagrams. The author discusses how mathematical techniques can be analyzed and explored for designing the algorithms of the future. Another advantage is that with the use of such a system, current models can also be optimized.

In the field of explainable AI, Zahoor et al. [40] present the use of functional analysis diagrams, such as LIME, to evaluate machine learning models. The authors' application ensures the fairness, reliability, and correctness of models, thereby going beyond merely accuracy measures, which is especially valuable in the case of text classification and sentiment analysis.

Prado and Digiampietri [133] work on the contribution of functional analysis diagrams towards automating some of the functionalities of the feature engineering process. This review establishes the ability of such diagrams to automate the process in a machine learning pipeline, hence making feature selection an efficient process.

Yang [41] clearly recognizes how such diagrams of functional analysis lead to exploring and mastering several machine learning algorithms for further ML model application on AI with structured and visually represented algorithms.

In another article by Sina Mohseni [134], functional analysis diagrams are said to enhance transparency and interpretability of machine learning systems through the development of a graphical representation of the decisions and predictions made. Toward the goal of making interpretable and explainable AI, these diagrams serve similar objectives.

In conclusion, functional analysis diagrams are expected to play an important role in pushing the frontier of machine learning and artificial intelligence applications because of the clear, transparent, and structured representation of complex systems and algorithms. The result of this will be not just toward the analysis and optimization of models but also towards the general objectives of explainability and transparency in AI.

3.3 Method

3.3.1 Addressing the Research Challenges

Table 3.1: Evaluation of Techniques for Addressing Challenge Characteristics

Characteristic	**SBD**	**FA**	**TRIZ**	**TRIZ-DSS**
Diversity & Novelty	C	C	C	C
Fairness	C	C	C	C
Long-Tail	C	C	C	C
Cold Start	C	C	C	C
Ranking & Relevancy	C	C	C	C
Reproducibility	C	C	C	C
Overfitting	C	C	C	C

P: Problem Addressed - The technique directly impacts and addresses the challenge.
C: Problem Mitigated - The technique partially impacts and mitigates the challenge.
N: No Direct Impact - The technique does not significantly impact the challenge.

This PhD thesis approach is characterized by systematic steps towards optimization of the CFRS tasks, which rely on further data representation and innovative mechanisms for further improvement of the system performance. As shown in Table 3.1, the selection criteria for the evaluation framework are also included, as it would pave the way for a comprehensive and effective evaluation of the methodologies proposed. A structured approach is going to offer a comprehensive and pragmatic solution for optimization of a CFRS.

3.3.2 Comparative Analysis of Analytical Methodologies

This subsection provides an in-depth comparison of various analytical methodologies for CFRSs, as outlined in Table 3.2. Each methodology is examined against critical characteristics to identify its applicability and effectiveness in the analysis and development of CFRSs.

SBD (System Block Diagrams) & FA (Functional Analysis) SBD visualizes dependencies and interactions and running analysis of the impact of change, scoring an advanced (A) level across these areas. Their strength in granularity of analysis is what makes them particularly effective in the decomposition of complex systems into manageable parts, giving clear insight into system functions and their interrelations.

UML (Unified Modeling Language) UML [125] is a powerful mechanism for system architecture visualization and interaction. This happens with the I score in dependency and interaction visualization. These are versatile tools for modeling software-based systems, and more particularly, CFRSs, which offer clarity in understanding through diagrams.

Systems Dynamics Focusing on dynamic behavior modeling and risk and reliability assessment, Systems Dynamics is marked as advanced (A) in these areas. This methodology is adept at simulating system behaviors over time, making it invaluable for analyzing feedback loops and time-dependent phenomena in CFRSs.

ABM (Agent-Based Modeling) ABM is able to effectively simulate dynamic behaviors and has been rated at the intermediate level (I) for granularity analysis and change impact analysis. The model also helps in providing insights into the source of emergent system behaviors from a bottom-up approach that is well suited when the complex nature of interactions from CFRSs is considered.

State Transition This is the most detailed methodology for understanding system states and transitions. It receives I-scores for most categories, but it is particularly useful for mapping out possible states and transitions within a CFRS, and therefore possible identification of failure modes.

Petri Nets Petri Nets work very effectively in modeling dependencies and interaction, conducting analysis of change impact. With their power in modeling concurrent processes and events, they provide a good framework to analyze complex CFRSs.

DSM (Design Structure Matrix) DSM is excellent in representing dependencies and interactions and performing impact analysis of change, mirroring the strengths of SBD & FA. With the potential to map the elements a system is composed of and their linkages, it has powerful diagnosis and innovation capabilities in CFRSs.

Given the comparative analysis in Table 3.2, **SBD & FA** emerge as comprehensive methodologies for analyzing CFRSs. Their advanced capabilities in granularity of analysis, dependency and interaction visualization, and change impact analysis equip developers with a profound understanding of system architecture and functional dynamics. These methodologies provide the scaffolding for systematic deconstruction and analysis of CFRSs, ensuring thorough exploration of system components, their interactions, and the implications of system modifications. As such, for endeavors requiring deep system analysis and insight into component interdependencies, SBD and FA offer a substantiated, detail-oriented approach that aligns with the complex nature of modern CFRSs.

3.3.3 Comparative Analysis of Methodologies for Optimizing CFRSs

In the context of CFRSs, the choice of methodology is in effort to satisfy the particular system requirements and challenges. A comparison of methodologies which are good for the current challenges related to CFRSs is shown in Table 3.3 against a number of key characteristics of CFRSs. We now proceed to discuss each methodology by making reference to the evaluation in the table and conclude with the rationale for the choice of the Multiview approach.

Table 3.2: Expanded Comparison of Methodologies for Analyzing Recommender Systems

Characteristic	**SBD & FA**	**UML**	**Systems Dynamics**	**ABM**	**State Transition**	**Petri Nets**	**DSM**
Granularity of Analysis	A	I	B	I	I	I	B
Dependency and Interaction Visualization	A	A	B	B	I	A	A
Dynamic Behavior Modeling	B	B	A	A	B	B	B
Failure Mode Identification	I	B	B	B	I	I	B
Risk and Reliability Assessment	I	B	A	B	B	I	B
Change Impact Analysis	A	I	B	B	I	A	A

A: Advanced - Represents the highest level of functionality, usually including comprehensive guidelines and methods.
I: Intermediate - Denotes a more detailed, process-oriented level of functionality.
B: Basic - Indicates a basic level of functionality.

CRISP-DM is a structured data mining process model that has worldwide support and a large community [135]. It scores low to moderate in most dimensions, ensuring its reliability and showing limited adaptability and innovativeness in a CFRS.

MLOps is the area responsible for handling the ML model life cycle with respect to their scalability, security, and operational efficiency [136]. The system presents great performance in factors crucial for the implementation and maintenance of scalable and secure CFRSs but shows intermediate adaptability to evolving user preferences.

DataOps applies Agile principles to data management with maximal speed and precision in analytics [137]. It greatly facilitates scalability and system integration. This, therefore, provides a very stable base for any CFRSs that require data handling and real-time processing capabilities.

ModelOps generalizes operational practices into the management of AI models, but with a special focus on scalability and security [138]. It would have a hard time handling the distinctive dynamics of CFRSs even when comprehensive support is given to operational concerns.

FSDL FSDL is intended to develop deep learning projects from end to end, and it gives structured ways of developing sophisticated models. It leans to technical implementation over CF-specifics, with characteristics like real-time adaptation and community engagement.

Agile ML Agile ML has really borrowed Agile development principles for ML projects, encouraging fast iteration and flexibility [139]. It was rated high for real-time recommendation and innovation in dynamic CF environments, but it did not perform as well as Multiview in handling cold-start and long-tail issues.

Comprehensive Methodological Framework for Collaborative Filtering (CMCF) our proposed solution integrates systems block diagrams, functional analysis, and TRIZ into a robust and versatile approach to system development. This is achieved here by integrating the macro-level insights of the systems block diagrams, which indicate the general architecture and interconnections among the parts of the system, with the micro-level detail, which is the functionality analysis focused on the individual functionalities and efficiencies of each component. TRIZ provides the ability to find innovative, effective, and efficient solutions to complex and contradiction-riddled problems of a system without deteriorating other aspects of it. In some way, then, these methodologies make sure that not only are CFRSs properly and explicitly designed but also that they be flexible, efficient, and continuously improved. This is an integrated strategic tool that made it possible to draw deep insight into both structural and functional aspects of the system, to innovate in combination with performance optimization and, at the same time, to be absolutely indispensable within the modern sphere of CFRS development.

As can be seen from Table 3.3, CMCF is the most comprehensive methodology for CFRSs. Advanced ratings in structured approach, flexibility, cold-start/long tail, and complete user and data privacy and security make it ideally well-suited for the complex landscape of CF. Such advanced ratings of methods are warranted due to its comprehensive structured approach, flexibility, cold-start/long tail, and full privacy and security of data and users, making it ideally well-suited for the advanced landscape of CF. Leveraging a fully integrated view of diverse perspectives and data sources, Multiview addresses the nuanced challenges of CF directly, from the improved accuracy of the recommendations to more scalable and secure operations. Therefore, for CFRSs requiring a balance of innovation, adaptability, and user-centric solutions, Multiview offers a substantiated framework that aligns with the multifaceted demands of modern CFRSs.

Table 3.3: Comparison of Methodologies for CFRSs

Characteristic	**CMCF**	**CRISP-DM**	**MLOps**	**DataOps**	**ModelOps**	**FSDL**	**Agile ML**
Structured Approach	A	A	A	A	A	A	A
Adaptability to CF	A	B	B	B	B	I	I
Real-time Recommendations	B	B	I	B	I	I	A
Evaluation Standards Integration	A	B	I	B	B	B	B
Innovative Solutions	B	B	I	B	B	B	A
Handling Cold-Start/Long-Tail	A	B	B	I	I	B	I
Scalability	B	B	A	A	A	A	B
User Privacy and Data Security	B	B	A	A	A	B	B
Adaptation to Evolving User Preferences	A	I	B	B	I	I	A
Integration with Existing Systems	B	I	A	A	A	B	B
Community and Support	B	A	A	B	B	B	I

A: Advanced - Indicates the methodology is well-suited to address the characteristic with comprehensive support.
B: Basic - Shows the methodology provides some support but may not fully address the characteristic's complexities.
I: Intermediate - Denotes the methodology offers a moderate level of support, balancing between basic and advanced.

3.4 Application of TRIZ-Enhanced DSS to Improve CFRSs

In the area of CFRSs, the incorporation of DSSs with the principles of the Theory of Inventive Problem Solving (TRIZ) will bring about altogether new enhancements to CFRS techniques. The proposed DSS Implementation involves rule-based decision-making along with clustering for grouping users and items.

To illustrate, TRIZ-Enhanced Collaborative Filtering algorithm is designed to bridge specific inherent issues in the existing CFRSs, such as data sparsity, decay in user interests with time, and the personalization-scalability tradeoff.

Both Algorithm 1 and Figure 3.1 are developed within a structured framework incorporating data normalization, strategic user and item categorization, and data enhancement techniques in the preparation of the dataset for effective model training. It is hoped that the inclusion of such an algorithm will help detect and resolve problems in CFRSs, optimize trade-offs between clashing system requirements by applying TRIZ principles. It not only makes the recommendations precise and fitting, but also makes the system overall adaptable and efficient.

This paves the way for a detailed exploration of the algorithm's components, steps in its implementation, and the expected benefits for CFRSs.

3.4.1 Data Preprocessing

Preprocessing may include rating normalization, time decay adjustments, and outlier management. A dataset is partitioned into a training set and test set. In the process of feature engineering, new attributes can be

Algorithm 1 DSS Implementation for TRIZ-Enhanced Collaborative Filtering (DSS-TRIZ-CF)

Input: Dataset D, decay rate λ, batch size S, number of epochs E, embedding dimensions $\boldsymbol{E}_u, \boldsymbol{E}_i$, threshold
Output: Trained model $\mathcal{M}_{TRIZ}$

Step 1: Normalize and adjust ratings for decay.
Step 2: Categorize users and items based on median values.
Step 3: Enhance data by duplicating instances with low ratings.
Step 4: Set up DSS models rooted in TRIZ principles.
Step 5: Train models and evaluate them using metrics like MSE, RMSE, and MAE.
Step 6: Conduct statistical analysis via paired t-tests.
Step 7: Endorse principles that demonstrate notable improvement through DSS.
return $\mathcal{M}_{TRIZ}$ alongside recommendations.

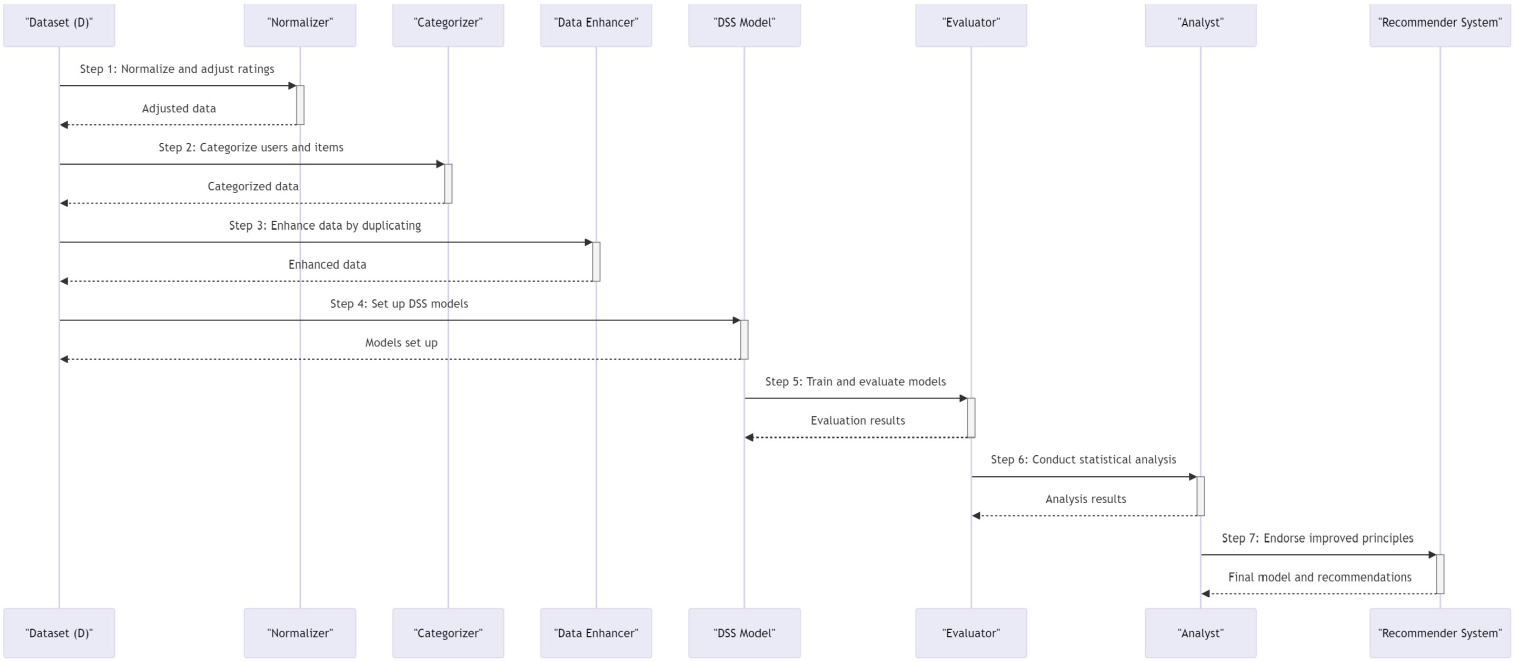


Figure 3.1: Systems block diagram of TRIZ-Enhanced CFRS Implementation.

worked out using the available dataset.

3.4.2 Implementing TRIZ Principles in CFRS

TRIZ, which stands for Theory of Inventive Problem Solving, is a method for understanding and solving problems. By applying these principles to Collaborative Filtering, the process of generating the recommendation is further refined to allow it to be more successful. A detailed description of the principles applied in CFRSs is given below.

Prior Action: Binarization of Ratings

The Prior Action principle seeks to do preparatory actions in advance of having to actually do the desired actions. In this context, by binarizing the ratings with respect to collaborative filtering, it is a form of preliminary action that greatly simplifies all subsequent computations and modeling steps.

The explicit feedback can be converted into implicit feedback so that the task of recommendation becomes easier and one has only to predict whether or not a user might interact with an item, rather than predicting the exact rating of an item to be given by the user. This is a pre-emptive step that might aid in dealing with sparse

data and also reduce the complexity of the recommendation task.

$$r' = \begin{cases} 1 & \text{if } r \geq \theta \\ 0 & \text{otherwise} \end{cases}$$

Here, r' is the binarized rating, r is the original rating, and θ is a predefined threshold, typically chosen based on the distribution of ratings in the dataset. This approach facilitates models to leverage user-item interactions and is particularly useful for datasets where explicit ratings are sparse or are not directly utilizable in modeling.

Local Quality

This principle focuses on recognizing and leveraging the inherent qualities of individual items. By counting the number of interactions for each item, we can segment items into popular and unpopular groups:

$$N_i$$

This distinction allows the model to treat popular items differently from niche items, catering to both mainstream and unique tastes.

Taking Out: Clustering of Ratings

The principle of "Taking Out" involves segregating or extracting a part of an object (or system) that can be treated independently to address a problem or provide improvements. In the context of collaborative filtering, the users and items can be clustered based on their rating behaviors, effectively "taking out" groups that exhibit similar patterns and addressing them distinctly.

$$\text{Cluster}_u = f_{\text{clustering}}(R_u)$$

Here, Cluster_u is the cluster to which user u is assigned, R_u represents the set of ratings given by user u, and $f_{\text{clustering}}$ is a clustering function, which may be implemented using various algorithms like k-means clustering, hierarchical clustering, etc.

Universality: Weighting Clusters

The principle of "Universality" pertains to making an object perform multiple functions, eliminating the need for other objects. By assigning weights to the clusters, the recommendation model becomes capable of addressing the varied preferences and behaviors embedded within each cluster, providing a "universal" framework that accommodates diverse user and item interactions.

$$w_c = f_{\text{weighting}}(\text{Cluster}_c)$$

Where w_c is the weight assigned to cluster c and $f_{\text{weighting}}$ is a function determining the weight based on the properties or impact of cluster c.

Through these weighted clusters, the model can provide recommendations that are sensitive to the nuanced preferences and behaviors of different user and item subsets, enhancing its overall effectiveness and relevance across the user-item space.

3.4.3 Model Training and Evaluation

Training and evaluating the model is a vital phase in a CFRS framework, especially after the application of TRIZ principles. This section provides insights into the methodologies employed in this process.

Model Architecture

The architecture for $\mathcal{M}$ could consist of a deep neural network with multiple layers. This network consumes the embedding vectors v_u and v_i as inputs, and after a series of transformations, it outputs a prediction for the rating.

- **Input Layer:** Concatenation of the embeddings for users and items followed by network processing.
- **Hidden Layers:** Several dense layers with activation functions like ReLU to introduce non-linearity.
- **Output Layer:** The output is a single neuron with an activation function like sigmoid, scaled to the range of normalized ratings.

Training Process

The model is trained using a supervised learning paradigm:

1. Data set D is split into a training and validation set.
2. An optimizer, such as Adam or RMSProp, is chosen to minimize the loss function L.
3. Batch training is conducted over many epochs, where mini-batches of data are forward propagated through the network and the gradients back propagated to update the weights.
4. Performance on the validation set is monitored to prevent overfitting. Techniques like early stopping are used.

Evaluation Metrics

While the mean squared error L is a primary metric, other metrics might also be valuable, such as:

- **Root Mean Squared Error (RMSE):** Provides an interpretable value about the average error in the original rating scale.
- **Mean Absolute Error (MAE):** Another error metric that gives the average of absolute prediction errors.
- **R-squared (Coefficient of Determination):** Indicates what proportion of variance in the dependent variable, ratings, can be explained by the independent variables, embeddings.

Comparative Analysis

Whether the TRIZ-enhanced CFRS model $\mathcal{M}_{TRIZ}$ is effective can be determined by comparison to a baseline, $\mathcal{M}_{base}$. The baseline model is typically a standard CFRS model without TRIZ modifications. Comparative metrics then indicate performance gains.

Hyperparameter Tuning

With the complexity and the multitude of layers in $\mathcal{M}$, hyper-parameters like learning rate, batch size, and the number of hidden neurons are crucial. Optimization is conducted using grid search or random search techniques to find the best hyper-parameters that minimize the loss L on the validation set.

3.4.4 Statistical Analysis

The paired t-test is a statistical central point in the analysis for determining the effectiveness of a model by comparing it to a baseline model. The assumptions of normality of data and absence of outliers are accorded proper attention. The interpretation of results focuses on both statistical relevance and real-world applicability.

3.5 Conclusions

This chapter explains a new method for evaluating and improving recommenders, focusing on E-commerce-based CFRSs. It applies TRIZ together with System Block Diagrams and Functional Analysis, specifically designed to deal with the weaknesses of existing approaches, like biases and limited applicability. The method is part of a multidisciplinary framework meant to enhance CFRSs.

The techniques in the TRIZ-Enhanced DSS for CFRSs involve rule-based decision-making along with clustering for grouping users and items. Deterministic weighting is applied to balance preferences. Processes like rating normalization and temporal decay adjustments maintain data consistency. Binarization is used to simplify feedback. Functional analysis is employed to organize the system. Optimization heuristics manage trade-offs, and paired t-tests are used for validating performance.

We suggest integrating TRIZ with CFRS into a DSS to address the issues faced by recommendation systems. This combination encourages innovation in data-driven applications and helps create reliable CFRSs. Using advanced mathematical and algorithmic approaches, it aims to enhance recommendation accuracy and improve decision-making.

Chapter 4

Solution 1 - An ISO-based Temporal Graph Collaborative Filtering Recommender System

With reference to sections 1.4 and 1.6, this chapter presents an innovative technique in the optimization of CFRSs by hybridizing ISO standards with temporal graph-based approaches. Following this CFRS design methodology will improve evaluation and align with objectives in section 1.7.

Within this chapter context, we highlight the basics and significance of matrix representation in dealing with the GCNs, leading to an in-depth explanation of GCNs. A sketch of prominent characteristics of the proposed solution, dealing with the construction and normalization of feature matrices with the application of a controlled propagation mechanism and setting up an appropriate evaluation framework, is drawn. The following sections will detail the implementation and experimental analysis, with a special focus on performance metrics, model evaluation, and implications for improving accuracy, diversity, and confidence in a CFRS.

4.1 Introduction

Graph Convolutional Networks (GCNs) is a neural network architecture that are designed to operate on graph-structured data and help in the learning of representations of the graph nodes to encode both features of these nodes and learn patterns of connections. GCNs are powerful for problems where relationships or interactions are explicitly mapped, for instance, social network graphs, molecular structures, or recommendation systems.

4.1.1 Matrix Representation of GCNs

The operation of each layer in a standard GCN can be broken down into two key steps: the process of aggregation and combination, which can be modeled efficiently in matrix form.

Aggregation in Matrix Form

Aggregation is carried out using the adjacency matrix of the graph, $\mathbf{A}$, which along with a degree matrix,$\mathbf{D}$, is normalized in order to align the adjacency matrix for proper feature vector scaling during aggregation.

The aggregated matrix $\mathbf{A}'$ is often computed as follows:

$$\mathbf{A}' = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$$

This normalized adjacency matrix is then used to perform the aggregation:

$$\mathbf{A}'\mathbf{H}^{(l-1)}$$

where $\mathbf{H}^{(l-1)}$ is the matrix of features for all nodes at layer $l-1$.

Combination in Matrix Form

Combination involves updating the node features by applying a transformation matrix $\mathbf{W}^{(l)}$ and a non-linear activation function σ. The new features at layer l are computed as:

$$\mathbf{H}^{(l)} = \sigma(\mathbf{A}'\mathbf{H}^{(l-1)}\mathbf{W}^{(l)})$$

where $\mathbf{W}^{(l)}$ is the weight matrix at layer l, which transforms the aggregated features into new features.

Complete GCN Layer Operation

Putting it all together, a single layer of a GCN in matrix form can be described as:

$$\mathbf{H}^{(l)} = \sigma\left(\mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}\mathbf{H}^{(l-1)}\mathbf{W}^{(l)}\right)$$

This formula represents how each node's features at layer l are a function of its own features and those of its neighbors at the previous layer $l-1$, transformed by the weights $\mathbf{W}^{(l)}$ and passed through a non-linear function σ (such as ReLU).

These matrix operations enable efficient computation of the feature updates for all nodes in the graph simultaneously, making GCN suitable for large-scale graph data. This efficiency is crucial for practical applications involving large networks, such as social media analysis, protein interaction networks, and large CFRSs.

4.2 Related Work

The literature reviewed has resulted in significant contributions related to current evaluation practices, limitations, and proposals for new and improved qualities of the evaluations for higher quality, reliability, and scalability of recommenders. In particular, ISO/IEC 25010:2011 is the focal point of the development of standards for the evaluation of software product quality, including recommenders, and this is a critical point in moving the field toward more structured and standardized evaluation frameworks.

Nuzula and Rochimah (2023) employed a descriptive quantitative research design to evaluate the quality of the Human Resource information systems using ISO/IEC 25010, specifically functional suitability and usability [117]. In another vein, Al Jurdi et al. (2022) postulated a neighborhood-based evaluation framework through which more detailed insights regarding the performance variance of the traditional RSs can be provided [118]. This shows that there is always a constant urge to examine alternative ways to deal with the failures of the existing evaluation process in order to get a more accurate perception of the system performance.

Henriques and Pinto (2023) proposed an evaluation mechanism using online and offline evaluation indexes, suggesting a holistic approach to the evaluation of RSs in multidimensional performance [140]. Trichkova-Kashamova (2021) applied the ISO/IEC 25010 quality model in order to decrease subjective influences in the estimation of quality and effectiveness of the software product. It was proven that the model is applicable to different kinds of information systems [116].

The literature also identifies gaps in the currently used methods of evaluation regarding universality, ease of implementation, and resource efficiency of the methods used [115]. Wang et al. [141] conducted a study that recommended the evaluation of the effectiveness of software implementation with Large Language Models, making the point that the methods applied to evaluation must be fitting for the process. In this review, we have realized the diversification of efforts and approaches in the scholarly community for improvements in evaluation methodologies for RSs. The community is currently focusing on handling the present limitations and considering standards, such as the ISO/IEC 25010:2011 standard, to move the field towards a more reliable, scalable, and quality-sensitive evaluation.

This further emphasizes the richness of current efforts and perspectives taken in academia for improving

the methodologies applied in the evaluation of RSs. This becomes, of course, a more reliable, scalable, and quality-driven progress in the evaluation practice if the current limitations are taken into account and standards like ISO/IEC 25010:2011 are absorbed.

4.3 Method

In this section context, we propose Temporal Graph Collaborative Filtering (TGCF).

4.3.1 Background

The development of graph-based recommendation systems has been a journey from Neural Graph Collaborative Filtering (NGCF) [61] to Light Graph Convolutional Network (LightGCN) [23] and, most recently, the published Simplifying Graph-based Collaborative Filtering for Recommendation (SGCF) [142]. Each system has brought new ideas on how to utilize the graph structures for making recommendations. In these recommendation models, the temporal dynamics and their integration are different in the frameworks, with a varying level of possible adaptability.

NGCF: Neural Graph Collaborative Filtering

NGCF introduces an innovative approach to graph-based collaborative filtering, meticulously engineered to exploit complex user-item interactions through high-order connectivities.

NGCF Equation:

The core operation of NGCF is encapsulated in the following propagation rule:

$$E^{(l)} = \text{LeakyReLU}\left((\hat{L}+I)E^{(l-1)}W_1^{(l-1)} + \hat{L}E^{(l-1)} \odot E^{(l-1)}W_2^{(l-1)}\right) \tag{4.1}$$

Symbols in this equation represent the following: $E^{(l)}$ is the embedding matrix at layer l, incorporating user and item features; $\hat{L}$ is the normalized Laplacian matrix, enabling the propagation of these features across the user-item graph; $W_1^{(l-1)}$ and $W_2^{(l-1)}$ are weight matrices for transforming the features; and I denotes the identity matrix.

The layered architecture of NGCF allows the fine-grained capture of collaborative filtering signals, but it does not deal with the diversity of recommendations or incorporate the temporal dynamics adequately within its algorithmic design.

LightGCN: Light Graph Convolutional Network

Since NGCF is overcomplicated by feature transformation layers and non-linear activation functions, LightGCN simplified the neural network learning process. LightGCN efficiently discards the significant complexity in NGCF, which results from complex layers of feature transformation and non-linear activation functions.

LightGCN Equation:

The essence of LightGCN's operational framework is captured in its embedding update rule:

$$E^{(k+1)} = \left(D^{-\frac{1}{2}}AD^{-\frac{1}{2}}\right)E^{(k)} \tag{4.2}$$

In this formulation, $E^{(k)}$ represents the embedding matrix after k iterations, A is the adjacency matrix reflecting user-item interactions, and D is the diagonal degree matrix, utilized here to normalize A, thereby ensuring the proportional propagation of features across the graph.

LightGCN further simplifies the NGCF model by sacrificing non-linear activations and feature transformation, making it more easily scalable to training with large datasets. Such a reduced model retains its accuracy, or it might even surpass NGCF by focusing only on the central idea: collaborative filtering. However, such a reductionist approach allows static representation of the formed user-item interactions and does not capture any temporal dynamics and diversity in recommendations. Both these limitations can easily be overcome by dynamic and diverse recommendation mechanisms, which might elevate LightGCN's recommendation effectiveness toward current complex needs.

SGCF: Simplifying Graph-based Collaborative Filtering for Recommendation

SGCF extends the efficiency ethos of LightGCN, but with an innovative incorporation of a graph-condensing technique to effectively manage computational challenges related to large-scale graphs. This strategy condenses the graph into a simple form that is easily handled during processing and learning within expansive networks.

The core mechanism of SGCF operates as follows:

$$E^{(k+1)} = SE^{(k)}W^{(k)} \tag{4.3}$$

Here, S represents the normalized adjacency matrix, including self-loops to maintain feature integrity; $E^{(k)}$ is the embedding matrix at the k^{th} iteration; and $W^{(k)}$ denotes the transformation matrix applied at each layer.

Graph condensation underpins the SGCF strategy to enhance computational efficiency through a reduction in graph complexity, thereby making it possible to deal with huge and intricate networks. Similar to its predecessors, SGCF does not present a complete answer to diversity in recommendations, as it does not address temporal aspects and popularity, which are the two most important aspects of diversity. Moreover, there is not much room for further development either in the design of its algorithm or within its experimental scope. These limitations make the model less capable of delivering dynamic, diverse, and contextually relevant recommendations.

4.3.2 The proposed Solution

TGCFs represent another step forward in the evolutionary development of graph-based collaborative filtering, deliberately designed to attempt to surpass the limitations of predecessors like NGCF, LightGCN, and SGCF. However, the dynamic and temporal nuances of user preferences and item relevancies are still not captured in previous models. In this work, such temporal features as the popularity and diversity of items has been innovatively introduced in the framework of graph convolutional networks. This would not only help enhance the adaptability of the model for temporal variations but also enhance the recommendation process via a better understanding of temporal dynamics. Temporal Graph Convolutional Networks set a new standard in the business of generating accurate, dynamic, and contextually rich recommendations by design, yielding a significant stride towards making recommendation systems more responsive and temporally aware. By using such advanced mechanics of adapting node influences, as detailed in the following subsections, it pushes that frontier even farther.

Temporal Features Integration (TFI)

We integrate the temporal features within the proposed model with the ability to capture the dynamics of item popularity and diversity over time, so that the model can adapt in an enhanced way to the temporal variation of user preferences and item relevancies.

Constructing the Feature Matrix

For each item i, we construct two vectors to encapsulate its dynamics across a 12-month period:

- **Temporal Popularity Vector** ($V^i_{\text{temp\_pop}}$): This is defined as $V^i_{\text{temp\_pop}} = [p^i_1, p^i_2, \ldots, p^i_{12}]$, where each p^i_j quantifies the popularity of item i in month j, such as the number of interactions or ratings. Formally, we calculate p^i_j as:

$$p^i_j = \frac{\text{interactions\_month}^i_j}{\sum_{k=1}^{12} \text{interactions\_month}^i_k}$$

- **Temporal Diversity Vector** ($V^i_{\text{temp\_div}}$): Similarly, this is described as $V^i_{\text{temp\_div}} = [d^i_1, d^i_2, \ldots, d^i_{12}]$, with each d^i_j measuring the diversity of interactions for item i in month j, for example, the variability or standard deviation in ratings. Formally, we calculate d^i_j as:

$$d^i_j = \sqrt{\frac{1}{N^i_j} \sum_{k=1}^{N^i_j} (r^i_k - \bar{r}^i_j)^2}$$

  where N^i_j is the number of ratings for item i in month j, r^i_k is the k-th rating, and $\bar{r}^i_j$ is the mean rating for item i in month j.

Together, these vectors form a comprehensive feature vector $F^i_{\text{temp\_features}}$ for each item, which, when aggregated across all items M, results in the feature matrix $F_{\text{temp\_features}}$ with dimensions $M \times 24$.

Normalization of Feature Vectors

Given the distinct nature of the temporal popularity and diversity metrics, each type of vector is normalized separately to maintain their individual scales and significance:

$$\hat{V}^i_{\text{temp\_pop}} = \frac{V^i_{\text{temp\_pop}} - \min(V^i_{\text{temp\_pop}})}{\max(V^i_{\text{temp\_pop}}) - \min(V^i_{\text{temp\_pop}})}$$

$$\hat{V}^i_{\text{temp\_div}} = \frac{V^i_{\text{temp\_div}} - \min(V^i_{\text{temp\_div}})}{\max(V^i_{\text{temp\_div}}) - \min(V^i_{\text{temp\_div}})}$$

This reflects how temporal popularity and temporal diversity are seen as complementary yet distinct. In the case of items, these features are infused more subtly in the TGCF model. Normalizing $V^i_{\text{temp\_pop}}$ and $V^i_{\text{temp\_div}}$ separately allows the model to better accommodate the unique contribution of each of the temporal metrics, which therefore carries superior predictive performance and diversity of recommendations.

Transformation Function (TF) f

The function f integrates the normalized temporal features ($\hat{F}_{\text{temp\_features}}$) into the initial item embeddings ($E^{(0)}_{items}$) through a linear transformation, reflecting the temporal dynamics in the item embeddings from the start:

$$E^{(0)}_{items} = \hat{F}_{\text{temp\_features}} \cdot W + b$$

where:

- W is a weight matrix of dimensions $24 \times d$, transforming the 24-dimensional feature vectors into the d-dimensional embedding space.
- b is a bias vector of dimensions $1 \times d$, applied directly to each row in the resulting matrix, enriching the initial embeddings with a bias adjustment across all d dimensions.

This method ensures that the initial embeddings $E^{(0)}_{items}$ for all items are effectively enriched with temporal insights, setting a foundation for the TGCF model to utilize these dynamics in generating more accurate and temporally relevant recommendations.

4.3.3 Controlled Propagation Mechanism (CPM)

Given the embedding matrix $E^{(k)}$ with dimensions $N \times d$, where N is the total number of nodes (users and items) and d is the embedding dimensionality, we introduce the control vector W_{control} (dimensions $N \times 1$) to dynamically adjust each node's influence during the propagation process. Each entry in W_{control} corresponds to a control weight for a specific node, aimed at modulating its contribution based on temporal relevance or other factors.

The modulation process begins by applying the sigmoid function to W_{control}, normalizing its values to the range 0 to 1:

$$\sigma(W_{\text{control}}) = \frac{1}{1 + e^{-W_{\text{control}}}}$$

This normalization step prepares W_{control} for influencing the propagation mechanism:

$$E^{(k+1)} = \tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}(E^{(k)} \odot (\sigma(W_{\text{control}})\mathbf{1}^T))$$

In this formula:

- $\sigma(W_{\text{control}})$ scales the control weights, adjusting the nodes' embedding influence proportionally.
- The operation $\odot$ represents element-wise multiplication, facilitated by broadcasting $\sigma(W_{\text{control}})$ across all embedding dimensions for each node, ensuring that the control weights uniformly scale the embeddings.

By integrating W_{control} through this controlled propagation mechanism, the TGCF model gains the ability to dynamically adjust to temporal dynamics and other contextual factors, enhancing both adaptability and the accuracy of recommendations.

4.3.4 Selecting a Quality Framework for Evaluating CFRSs

With a reference to Table 4.1, the evaluation of CFRSs needs the selection of a quality framework that takes care of a multitude of expectations covering algorithmic efficiency to user satisfaction. Among the currently existing, the ISO/IEC 25010:2011 is found to be bestowing many advantages in this connection because of its comprehensiveness towards product quality and quality in use. It doesn't solely focus on technical excellence but also provides a lot more emphasis on user-centered evaluation; hence, being the strategic choice by evaluators whose choice is an all-encompassing framework.

Whereas ISO/IEC 9126 and ISO/IEC 12207 give a quality perspective of foundational and lifecycle orientation respectively, they may not completely account for the specific requirements of CFRSs.

Frameworks, particularly CMMI and SPICE, were evidently process maturity- and process improvement-oriented, which in turn has generated invaluable input towards the effective design and maintenance of high-quality software systems. Nevertheless, the quality in use, which is a more user-centered metric and holds a very important role as far as the success of the CFRSs with end users is valued greatly.

Therefore, choosing ISO/IEC 25010:2011 for the evaluation of CFRSs is elaborate in itself because it also examines the technical excellence that is to be accorded to users. Table 4.1 shows that ISO/IEC 25010:2011 reflects higher level of applicability. Accordingly, this will enable the possibility to deeply investigate and improve CFRSs, making sure they will behave at their best and evoke positive user response.

Table 4.1: Comparison of Standards and Frameworks for Evaluating CFRSs

Evaluation Point	**ISO/IEC 9126**	**ISO/IEC 14598**	**ISO/IEC 25012**	**ISO/IEC 12207**	**CMMI**	**IEEE 1061**	**SPICE**	**ISO/IEC 25010:2011**
Data Quality Assessment	B	I	A	B	I	I	I	A
Algorithm Performance	B	I	B	I	I	A	I	A
Functionality and Reliability	A	A	B	A	A	B	A	A
Usability	A	I	B	I	I	B	I	A
Maintainability and Adaptability	A	A	B	A	A	I	A	A
Development Process Maturity	B	I	B	A	A	I	A	I
System Lifecycle Management	B	I	B	A	A	B	A	A
Quality Metrics Application	B	A	I	I	I	A	I	A
Portability	A	I	B	I	I	B	I	A
Risk Management	B	I	B	I	A	I	A	I
Objective Criteria for Evaluation	B	A	I	I	I	A	I	A
Process Improvement	B	A	B	A	A	I	A	I

B: Basic - Indicates a foundational or initial level of applicability.

I: Intermediate - Denotes a more detailed, process-oriented level of applicability.

A: Advanced - Represents the highest level of applicability, often including comprehensive guidelines and methodologies.

4.4 First Set of Experiments: Analysis of Performance Metrics

This is the first set of experiments that were conducted in the evaluation of its performance. In this section, implementation specifics are detailed, after which a thorough exposition of an array of performance metrics is detailed. The experiments are delved into in granularity, focusing on metrics, namely Shannon Entropy, recall performance, Normalized Discounted Cumulative Gain (NDCG), the Gini Index, popularity scores, item coverage, and loss metrics. Each metric is analyzed for patterns and themes, which have implications on and for the proposed Temporal Graph Collaborative Filtering (TGCF) system and on and for larger recommendation system frameworks. This means that the performance adopted in this comprehensive analysis brings forth a clear identification of strength areas and areas to improve in the proposed solution.

4.4.1 Implementation Details for Experiments

We enriched our experiments with the guidelines and recommendations of RecBole [143–146]. RecBole is an end-to-end comprehensive modular framework engineered for developing and experimenting with recommendation.

We have applied the model from this chapter within the scope [1] of the ML-100K dataset [60]. The dataset is divided into the training set and testing set: 80% of the data belong to the training set, while the rest belongs to the testing set. Also, each of the events in the training set is treated as a positive interaction and, at random, one of the negative interactions is selected from the training set. We have refined the ML-100K dataset to guarantee a fair comparison across all algorithms and to ensure an adequate number of interactions with niche items for all evaluated metrics, such as Shannon entropy. Details about the baselines used can be found in section 2.1.12, while the details about the used metrics can be found in the section 2.1.11.

[1] More information on ML-100K Dataset is found in section 2.1.12.

4.4.2 Analysis of Shannon Entropy Values

Table 4.2 reports Shannon entropy values calculated over different list lengths, from @5 to @100. Hence, Shannon entropy in this case most likely reflects the level of diversity in recommendations, where higher values mean more diversity. Table 1(in Appendix .1) shows the CFRSs' approach to handling the challenges described in Section 1.4.

Patterns and Themes

- **Comparison of Diversity:** The entropy values for both the models, SpectralCF and LDiffRec, are higher than for the other models for all lengths of the list. It indicates that the recommendations produced from these algorithms are diversifying to the maximum. This might be useful in cases where user satisfaction is based on varied and unexpected recommendations.

- **Performance of TGCF:** Our proposed solution, TGCF, presents significantly higher entropy values compared to most of the baseline models, especially at smaller list sizes (@5 to @20). Thus, TGCF is more effective in introducing diversity early in the recommendation list. It becomes crucial for user engagement in practical applications.

- **Low Diversity Baselines:** Models such as LightGCN, ENMF, and RecVAE all tend to have low entropy across most measures. This indicates that these models might inherently be conservative in their recommendation strategies, possibly having a tendency to select only popular or highly rated items.

- **Mid-Range Performers:** Models such as SimGCL and XSimGCL fall under medium diversity—far beyond the low diversity baselines but still far below the top models or TGCF. While this kind of placement can serve as a compromise between recommending highly popular items and introducing highly novel ones, at the same time, it does not stand out in the direction of one of the two extremes.

- **Decline with List Size:** For all but one model, entropy declines with list size, which is intuitive. The largest lists tend to draw from the largest set of recommendations, which contains one copy of highly diverse top recommendations, and so bring in more novel items in a gradual increase of repetition or in less novel items.

Implications for TGCF

Implications are represented in High entropy values in TGCF, especially visible at small list sizes, highlight its strength in drawing up diverse recommendations right from the beginning. It can, therefore, be very useful for user satisfaction through many different items being shown to the user, which catches their interest.

The data supports that TGCF is a strong candidate for a scenario where diversity is of prime importance. However, for use cases where "accuracy" also has a prime value, such as high-precision top-recommended items, one must look at the trade-offs from other models that show low diversity but can promise high precision.

4.4.3 Interpretation of Recall Performance Metrics

Table 4.3 provides recall performance metrics for various recommendation models at different list lengths (@5 to @100). Recall measures the model's ability to retrieve all relevant items within the top N recommendations, with higher values indicating better performance.

Patterns and Themes

- **Top Performer:** TGCF, our proposed solution, consistently outperforms other models across all metrics, especially notable at higher list sizes (@50 to @100). This indicates a strong ability of TGCF to capture most relevant items as the list size increases, which is crucial for user satisfaction in recommendation scenarios.

- **Lower Performance Models:** Models such as SpectralCF and LDiffRec demonstrate significantly lower recall rates across all list sizes. These models may struggle to capture all relevant items, potentially less suitable for applications where missing relevant recommendations could lead to user dissatisfaction.
- **General Improvement with List Size:** There is a general trend where all models show improved recall as list sizes increase. This is expected as a larger list size allows more opportunities for the model to recommend relevant items.
- **Mid-Range and Variable Performers:** Models like SimpleX, SGL, and SimGCL exhibit moderate recall values, with some variability across different list sizes. These models could be useful in more niche scenarios or might benefit from further optimization to enhance their recall.

Implications for TGCF

TGCF's superior recall performance, particularly at larger list sizes, highlights its robustness in ensuring that most relevant items are presented to the user, reducing the likelihood of missed recommendations. This feature makes TGCF a compelling choice for platforms where user engagement and satisfaction are directly tied to the comprehensiveness of the recommendation list.

The data shows TGCF is a suitable choice for scenarios where diversity is important. For cases where accuracy is also needed, such as top recommendations requiring high precision, other models with lower diversity but better precision should be considered.

4.4.4 Interpretation of Normalized Discounted Cumulative Gain (NDCG)

Table 4.4 reports NDCG values of different recommendation models at different list lengths (@5 to @100). An NDCG measures how well the ranking represents the relative relevance of the different recommendations in the list: High values point to a nice rank of the most relevant items towards the top of the recommendation list.

Patterns and Themes

- **Strongest Performance:** TGCF has the highest NDCG values with all the evaluation metrics. Such results point to an exceptional performance in ranking effective and relevant items, regardless of the list sizes. It is particularly good with the longer lists, showing a sustained effectiveness as the recommendation list lengthens.
- **Consistently High Performers:** Other model that turn out to perform quite well at most list sizes is RecVAE, having relatively high scores toward the larger list sizes. In this respect, what makes RecVAE very important is that it preserves ranking quality when the list size is expanded.
- **Low Performing Models:** SpectralCF and LDiffRec are amongst the models with the lowest NDCG scores. These indicate there could be trouble in making the most relevant placements on top of the recommendations. Thus, these models might be effective for recommendation contexts but with a low effect unless modified for good performance.
- **General Trend of Increase:** Most of the models show that with increased list size, NDCG also increases; that is, although starting with a worst beginning, the model might recover relevance with increased items among the recommendations.
- **Mid-Range Performers:** Models like LightGCN, SimGCL, and SGL lie in the mid-range of NDCG values. They are mid-ranged in their performance, performing well but not as strongly as top models. Such would be the models of choice in application cases where one needs to balance performance with computational efficiency.

Implications for Recommendation Systems

The general results, measured from the NDCG score, reveal that some models rank pretty uniformly in getting good quality rankings, while other models have to be optimized in their ranking effectiveness. In this line, the model's ability to maintain the high values of NDCG across the lists' lengths is critical for some applications in which the users may explore deep into the recommendation lists. The selection of a model, for example, TGCF, in contexts that require high precision at the outset and relevance over time, should make the users more satisfied with its use. On the other hand, the use of models with lower precision would be more acceptable in the case that the user did not rely on the recommendation's precision, or in the case in which there are limitations on the system's resources.

4.4.5 Interpretation of the Gini Index

The Gini Index values in Table 4.5 for different recommendation models and a different list length (@5 to @100) can be assumed to indicate how the Gini Index, being an inequality measure, quantifies the level of inequality in the exposure of items or their distribution in the list of recommendations. The smaller the Gini Index, the better the distribution of the recommendations across items, which is key for not letting very few items take the bulk of recommendations and for a diverse pool of recommendations. Table 1(in Appendix .1) shows the experiment's approach to handling the challenges described in Section 1.4.

Patterns and Themes

- **High Inequality Models:** The SpectralCF and LDiffRec models present very high Gini Index values. High values of the Gini Index would mean high levels of inequality of item exposure. So, these models would recommend only a few items, probably making the high variety of choices less exposed to users.
- **Moderate Inequality Models:** Models such as DiffRec, ENMF, and XSimGCL have moderate Gini Index values in most cases, which indicate that the distribution is better but unequal compared to other models. This may be in the middle of personalization and diversity; hence, promising their usefulness in scenarios where both personalization and diversity are important.
- **Lower Inequality - TGCF:** The density plot showing the Gini Index values for the TGCF are lower; this is especially the case with longer list lengths. This means that it will give a more equitable spread of recommendations across different items, which can be really important in some use cases where the exposure to the wide variety of items is critical.
- **General Trend of Decrease:** In most cases, the general trend that Gini Index values decrease when list lengths are increased means that a model is more likely to provide fairer recommendations with a greater number of items within a list, which is, in fact, a greater chance to put in more items with the growing list length.
- **Performance Across Models:** Some models show a gradual and consistent decrease in the Gini Index as list length increases (e.g., LightGCN, RecVAE), while others start with extremely high values that decrease slowly (e.g., SpectralCF, LDiffRec). This could reflect different underlying algorithms' focus on either maintaining diversity or concentrating on a few highly relevant items.

Implications for Recommendation Systems

The Gini Index data across models reflects varied performance in terms of item distribution equity. For this interest in reducing item-exposure inequality, models with lower Gini Indexes, such as TGCF, would be more appropriate. In contexts where the priority is on maximal relevance for a few items rather than broad visibility, models with higher Gini Indexes might actually be useful.

4.4.6 Analysis of Popularity Scores Across Recommendation Models

Table 4.6 provides an overview of the popularity scores of the recommended items by different recommendation algorithms for different list lengths (@5 up to @100). These scores most probably reflect the average popularity of the recommended items for each model, with those having higher values meaning there is a propensity to recommend more popular items.

Patterns and Themes

- **High Popularity Bias:** The models SpectralCF and LDiffRec are highly biased towards promoting items that have high popularity scores. These models would not, therefore, serve well for those who seek novelty or look for niche content but would serve well for users looking for mainstream items. Table 1(in Appendix .1) shows the experiment's approach to handling the challenges described in Section 1.4.
- **Moderate Popularity Bias:** Some models include a moderate bias toward popularity, such as DiffRec, ENMF, and SimGCL. These models make recommendations that balance popular items with a variety of other items, hence the recommendation of a variety of content that may appeal to a wider audience.
- **Low Popularity Bias:** TGCF holds for the lowest popularity scores across all but the largest list sizes. This implies that it is more likely to derive a less popular and thus more diverse item, important for the user's long-tail content discovery.
- **Decrease with List Size:** Most models show a decreasing trend of their popularity scores with increasing list size. The decreasing trend reveals that small lists mostly contain very popular items and adding more items introduces unpopular choices—content diversity.

Implications for Recommendation Systems

The analysis of popularity scores can thus help select the recommendation models according to the desired balance of popular and niche content exposure. In environments that rely on the level of engagement of users in popular content, models with more popular scores may fare better. On the other hand, in settings where the discovery of less mainstream items is valued, models like TGCF may be more appropriate. It provides a critical understanding of the dynamics involved, which are crucial for deploying the recommendation system in alignment with strategic goals, either to maximize user satisfaction, content variety, or to promote lesser-known items.

4.4.7 Analysis of Item Coverage Across Recommendation Models

Table 4.7 below shows the item coverage for various recommendation models at various cutoffs (@5 to @100). Item coverage is the fraction of the total item catalog that receives at least one recommendation over the different recommended lists of length. The higher the value, the better the model can give a variety of items in recommendations. Table 1(in Appendix .1) shows the experiment's approach to handling the challenges described in Section 1.4.

Patterns and Themes

- **High Coverage Models:** The two models, namely TGCF and LightGCN, show the highest item coverage at all cutoff points, implying that these models recommend a high diversity of items from the catalog. This suggests the effectiveness of these models in surfacing a user to a large amount of different items—very important in platforms that encourage discovery.
- **Moderate Coverage Models:** The ENMF, HMLET, and RecVAE models also show good item coverage, although a bit lower than the top performers. These models balance the recommend popular items and assure the not-so-popular items have a fair share in the recommendation lists.

- **Low Coverage Models:** The LDiffRec and SpectralCF models have low coverage, signifying that the models are expected to give a narrow set of repeated recommendations. This might lead to only a small part of the available catalog being exposed to a user. This will be less satisfactory for users in environments where discovery is the priority.
- **Improvement with Increased Cutoffs:** The increasing general trend across all the models is that with an increase in the list size, there is an increase in the coverage of items—an indication that more items get the chance to be recommended in the bigger list sizes. This trend suggests the significant importance of list size in determining the extent of recommendations.

Implications for Recommendation Systems

Understanding item coverage is essential for choosing the right recommendation model based on the strategic goals of a platform. For instances where promoting a wide variety of items is critical, selecting models with high item coverage like TGCF or LightGCN would be advantageous. Conversely, in scenarios where a more focused set of recommendations is desired, models with lower coverage might be appropriate. Item coverage metrics help in assessing the capability of models not only to recommend effectively but also to enhance user engagement through diverse content discovery.

4.4.8 Analysis of Loss Metrics Across Recommendation Models

Table 4.8 reports comparative loss metrics of MAE, RMSE, and Log Loss for different recommendation models. The metrics evaluate the quality of the prediction, in which a low value of the metric means a better performance of the model.

Patterns and Themes

- **Outliers in Loss Metrics:** LDiffRec and SpectralCF models have very high values of all the three metrics, meaning that the models are making very serious mistakes in prediction. Such high loss values may be associated with the overfitting of models, underfitting, or data mismanagement.
- **Low Error Models:** Models such as SimGCL, HMLET, and TGCF reflect really low values over all the measures and are indicative of high accuracy in their predictions. For example, TGCF maintains quite competitive MAE and RMSE, as well as a moderate Log Loss, suggesting a balanced performance in accurately predicting user preferences.
- **Performance Consistency:** A closer look at most of the models built shows a very consistent relationship between the two metrics MAE and RMSE, as their plots look almost similar, considering that these two metrics measure predictive error. For instance, the differences in log loss vary more, meaning that there is a deviation in the way different models handle uncertainty.
- **Comparative Performance:** This comparison reveals that the difference in the model's precision can vary tremendously from one model to another, meaning that some models perform well while others struggle, leading to comparatively high error rates. This variation underpins the importance of model choice based on the specific performance needs.

Implications for Model Selection

Choosing the right model is based on loss metrics and application needs. For tasks needing precision, models with low MAE and RMSE should be preferred. When probability estimates are more important, Log Loss is the metric to focus on.

Table 4.2: Shannon Entropy Values Across Different Recommendation Algorithms.

Model	@5	@10	@20	@30	@40	@50	@60	@70	@80	@90	@100
DiffRec	0.0181	0.0144	0.0112	0.0099	0.0088	0.0083	0.0078	0.0075	0.0072	0.0069	0.0065
ENMF	0.0135	0.0105	0.0084	0.0076	0.0069	0.0066	0.0063	0.0060	0.0059	0.0058	0.0056
HMLET	0.0146	0.0113	0.0087	0.0076	0.0069	0.0067	0.0065	0.0064	0.0062	0.0059	0.0055
LDiffRec	0.0733	0.0536	0.0372	0.0311	0.0277	0.0257	0.0245	0.0236	0.0223	0.0210	0.0168
LightGCN	0.0124	0.0097	0.0078	0.0069	0.0064	0.0062	0.0060	0.0060	0.0058	0.0058	0.0053
RecVAE	0.0130	0.0104	0.0083	0.0072	0.0068	0.0066	0.0065	0.0064	0.0063	0.0059	0.0054
SGL	0.0137	0.0106	0.0084	0.0074	0.0067	0.0066	0.0064	0.0062	0.0060	0.0057	0.0051
SimGCL	0.0188	0.0142	0.0113	0.0096	0.0086	0.0081	0.0079	0.0077	0.0075	0.0071	0.0063
SimpleX	0.0204	0.0142	0.0093	0.0073	0.0064	0.0061	0.0059	0.0058	0.0056	0.0054	0.0050
SpectralCF	0.0811	0.0636	0.0471	0.0383	0.0326	0.0304	0.0286	0.0271	0.0254	0.0241	0.0184
XSimGCL	0.0154	0.0121	0.0094	0.0083	0.0076	0.0072	0.0069	0.0067	0.0065	0.0063	0.0058
TGCF	0.0354	0.0265	0.0205	0.0157	0.0140	0.0127	0.0118	0.0114	0.0106	0.0094	0.0075

Table 4.3: Recall Performance Metrics of Various Recommendation Models.

Model	**@5**	**@10**	**@20**	**@30**	**@40**	**@50**	**@60**	**@70**	**@80**	**@90**	**@100**
DiffRec	0.1412	0.2230	0.3366	0.4169	0.4759	0.5247	0.5665	0.6016	0.6334	0.6591	0.6824
ENMF	0.1380	0.2176	0.3360	0.4149	0.4767	0.5263	0.5683	0.6012	0.6314	0.6553	0.6778
HMLET	0.1286	0.2150	0.3288	0.4112	0.4758	0.5325	0.5780	0.6112	0.6426	0.6713	0.6974
LDiffRec	0.0611	0.1072	0.1703	0.2071	0.2454	0.2635	0.2976	0.3134	0.3669	0.3980	0.4226
LightGCN	0.1311	0.2144	0.3299	0.4115	0.4786	0.5358	0.5808	0.6194	0.6516	0.6792	0.7025
RecVAE	0.1425	0.2317	0.3476	0.4340	0.5012	0.5518	0.5903	0.6269	0.6567	0.6810	0.7056
SGL	0.1285	0.2085	0.3239	0.4032	0.4704	0.5236	0.5637	0.6022	0.6386	0.6664	0.6896
SimGCL	0.1360	0.2139	0.3237	0.4043	0.4668	0.5191	0.5662	0.6042	0.6349	0.6638	0.6870
SimpleX	0.1179	0.1997	0.3134	0.3919	0.4579	0.5016	0.5410	0.5748	0.6015	0.6280	0.6497
SpectralCF	0.0572	0.1067	0.1639	0.2099	0.2414	0.2784	0.2982	0.3171	0.3738	0.3961	0.4139
XSimGCL	0.1388	0.2254	0.3398	0.4249	0.4897	0.5438	0.5913	0.6272	0.6575	0.6835	0.7075
TGCF	0.1485	0.2418	0.3627	0.4530	0.5219	0.5748	0.6143	0.6524	0.6858	0.7096	0.7341

4.5 Second Set of Experiments: Evaluating The CFRSs using an ISO/IEC 25010:2011 based Framework.

In the lively E-commerce domain, CFRSs are contributing toward a better user experience and sales. The evaluation of such systems, in line with the ISO/IEC 25010:2011 standards [35], makes the approach holistic and multi-dimensional. The standard propounds a wide range of guidelines for quality software, with a main focus on 'Product Quality' and 'Quality in Use'. This methodology incorporates the use of such standards, with an explanation on why certain metrics are chosen for a performance dimension. Due to space constraint in the Thesis, we only include this simplified methodology. Table 1(in Appendix .1) shows the experiment's approach to handling the challenges described in Section 1.4.

4.5.1 Normalization Equation

To ensure standardization across different scales, the normalization formula is applied:

$$\text{Normalized Score} = 1 + 4 \times \left(\frac{\text{Metric Value} - \text{Min Value}}{\text{Max Value} - \text{Min Value}} \right) \tag{4.4}$$

Table 4.4: Normalized Discounted Cumulative Gain (NDCG) Across Models.

Model	@5	@10	@20	@30	@40	@50	@60	@70	@80	@90	@100
DiffRec	0.4318	0.4053	0.4027	0.4066	0.4125	0.4199	0.4256	0.4336	0.4401	0.4480	0.4993
ENMF	0.4196	0.3941	0.3952	0.4005	0.4069	0.4143	0.4207	0.4288	0.4354	0.4426	0.4925
HMLET	0.4022	0.3894	0.3893	0.3934	0.4009	0.4094	0.4168	0.4250	0.4340	0.4419	0.4949
LDiffRec	0.2432	0.2297	0.2231	0.2222	0.2224	0.2255	0.2296	0.2344	0.2388	0.2450	0.2914
LightGCN	0.4089	0.3895	0.3909	0.3951	0.4030	0.4108	0.4192	0.4280	0.4366	0.4443	0.4989
RecVAE	0.4336	0.4154	0.4123	0.4159	0.4245	0.4330	0.4405	0.4473	0.4551	0.4615	0.5121
SGL	0.4070	0.3852	0.3836	0.3880	0.3945	0.4026	0.4110	0.4190	0.4267	0.4339	0.4886
SimGCL	0.4157	0.3938	0.3914	0.3959	0.4017	0.4081	0.4161	0.4235	0.4313	0.4390	0.4932
SimpleX	0.3924	0.3737	0.3704	0.3745	0.3794	0.3866	0.3946	0.4004	0.4065	0.4129	0.4607
SpectralCF	0.2157	0.2098	0.2053	0.2059	0.2109	0.2141	0.2178	0.2232	0.2298	0.2365	0.2802
XSimGCL	0.4289	0.4072	0.4069	0.4099	0.4182	0.4255	0.4336	0.4420	0.4498	0.4574	0.5111
TGCF	0.4557	0.4302	0.4167	0.4264	0.4537	0.4712	0.4708	0.4770	0.4923	0.4972	0.5549

Table 4.5: Gini Index Comparisons for Recommendation Algorithms.

Model	@5	@10	@20	@30	@40	@50	@60	@70	@80	@90	@100
DiffRec	0.9500	0.9318	0.9018	0.8789	0.8572	0.8372	0.8193	0.8023	0.7862	0.7710	0.7564
ENMF	0.9382	0.9180	0.8872	0.8609	0.8365	0.8147	0.7944	0.7759	0.7584	0.7416	0.7259
HMLET	0.9416	0.9201	0.8899	0.8654	0.8428	0.8233	0.8050	0.7883	0.7732	0.7592	0.7463
LDiffRec	0.9928	0.9876	0.9782	0.9702	0.9621	0.9540	0.9465	0.9387	0.9310	0.9236	0.9165
LightGCN	0.9219	0.8987	0.8686	0.8445	0.8225	0.8033	0.7868	0.7720	0.7581	0.7440	0.7312
RecVAE	0.9288	0.9108	0.8828	0.8607	0.8413	0.8242	0.8091	0.7950	0.7821	0.7699	0.7586
SGL	0.9162	0.8946	0.8654	0.8414	0.8196	0.7994	0.7817	0.7643	0.7484	0.7329	0.7185
SimGCL	0.9574	0.9400	0.9142	0.8929	0.8724	0.8536	0.8367	0.8209	0.8063	0.7928	0.7800
SimpleX	0.9582	0.9441	0.9176	0.8905	0.8642	0.8402	0.8171	0.7958	0.7759	0.7579	0.7489
SpectralCF	0.9939	0.9889	0.9798	0.9716	0.9637	0.9560	0.9484	0.9408	0.9334	0.9262	0.9191
XSimGCL	0.9481	0.9307	0.9036	0.8818	0.8614	0.8428	0.8261	0.8106	0.7955	0.7812	0.7677
TGCF	0.8915	0.8724	0.8376	0.8035	0.7924	0.7761	0.7635	0.7440	0.7416	0.7210	0.7067

For metrics where a lower value indicates better performance, the score is inverted:

$$\text{Inverted Score} = 6 - \text{Normalized Score} \tag{4.5}$$

4.5.2 Dimensions Calculation and Metric Selection

Each dimension and its corresponding metrics are chosen to reflect specific aspects of the ISO/IEC 25010:2011 [35] standard.

1) Accuracy and Predictive Performance

Metrics: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Log Loss

$$\text{Accuracy Score} = \frac{\text{MAE Score} + \text{RMSE Score} + \text{Log Loss Score}}{3} \tag{4.6}$$

Explanation and Rationale: Aligned with the 'Functional Suitability' and 'Functional Correctness' aspects of ISO/IEC 25010:2011 [35], these metrics are chosen for their effectiveness in measuring the precision and reliability of the CFRS in matching user preferences and interests.

Table 4.6: Popularity Scores of Items Recommended by Different Algorithms.

Model	**@5**	**@10**	**@20**	**@30**	**@40**	**@50**	**@60**	**@70**	**@80**	**@90**	**@100**
DiffRec	243	221	195	180	168	159	152	146	141	136	132
ENMF	235	214	190	175	163	154	147	141	136	132	128
HMLET	226	206	185	171	160	152	145	139	134	130	127
LDiffRec	314	291	265	244	230	220	211	204	197	191	185
LightGCN	212	194	175	163	154	146	141	136	131	128	124
RecVAE	221	206	185	172	163	155	149	144	139	135	132
SGL	196	183	167	156	147	140	134	129	125	121	117
SimGCL	243	221	196	181	170	161	154	148	144	139	136
SimpleX	235	220	200	183	170	160	152	145	139	134	129
SpectralCF	335	308	273	251	236	224	215	208	200	194	188
XSimGCL	236	216	193	178	167	159	151	146	141	137	133
TGCF	192	175	162	150	142	135	129	124	120	116	112

Table 4.7: Item Coverage for Various Cutoffs

Algorithm	@5	@10	@20	@30	@40	@50	@60	@70	@80	@90	@100
DiffRec	0.1539	0.2074	0.2858	0.3387	0.3922	0.4242	0.4569	0.4831	0.5116	0.5383	0.5740
ENMF	0.2145	0.2929	0.3928	0.4516	0.5080	0.5472	0.5835	0.6174	0.6447	0.6649	0.6815
HMLET	0.1973	0.2733	0.3779	0.4492	0.5086	0.5490	0.5805	0.6174	0.6411	0.6619	0.6863
LDiffRec	0.0226	0.0368	0.0618	0.0796	0.0945	0.1111	0.1260	0.1432	0.1586	0.1717	0.1830
LightGCN	0.2442	0.3298	0.4320	0.5045	0.5609	0.6025	0.6298	0.6542	0.6768	0.7047	0.7249
RecVae	0.2305	0.3030	0.4017	0.4729	0.5139	0.5490	0.5740	0.6025	0.6405	0.6667	0.6916
SGL	0.2276	0.3060	0.4029	0.4718	0.5306	0.5722	0.6162	0.6560	0.6946	0.7249	0.7534
SimGCL	0.1438	0.2050	0.2769	0.3399	0.3933	0.4367	0.4700	0.5009	0.5300	0.5627	0.5829
Simplex	0.1325	0.2026	0.3339	0.4492	0.5312	0.5924	0.6376	0.6791	0.7089	0.7356	0.7641
SpectralCF	0.0190	0.0297	0.0475	0.0636	0.0790	0.0939	0.1093	0.1236	0.1373	0.1521	0.1652
XSimGCL	0.1818	0.2478	0.3422	0.4011	0.4528	0.5068	0.5472	0.5764	0.6043	0.6292	0.6477
TGCF	0.2525	0.3427	0.4491	0.5237	0.5805	0.6265	0.6523	0.6800	0.7026	0.7292	0.7706

2) Ranking Quality

Metrics: Recall, Normalized Discounted Cumulative Gain (NDCG)

$$\text{Ranking Quality Score} = \frac{\text{Recall Score} + \text{NDCG Score}}{2} \tag{4.7}$$

Explanation and Rationale: Corresponding to the 'Usability' characteristic of ISO/IEC 25010:2011 [35], Recall and NDCG are selected for their ability to measure the relevance of the recommendations to the users, showing user satisfaction.

3) Diversity and Reliability

Metrics: Shannon Entropy, Item Coverage

$$\text{Diversity and Reliability Score} = \frac{\text{Shannon Entropy} + \text{Item Coverage}}{2} \tag{4.8}$$

Table 4.8: Comparative Loss Metrics (MAE, RMSE, Log Loss) for Recommendation Models.

Model	MAE	RMSE	Log Loss
DiffRec	0.0230	0.0966	0.0455
ENMF	0.0203	0.0987	0.0543
HMLET	0.0165	0.0984	0.0509
LDiffRec	2.3411	4.1531	19.3812
LightGCN	0.0217	0.0979	0.0592
RecVAE	0.1557	0.2101	0.2307
SGL	0.0235	0.0986	0.0720
SimGCL	0.0153	0.0983	0.0521
SimpleX	0.1075	0.1590	0.2399
SpectralCF	2.9143	4.2820	23.5091
XSimGCL	0.0238	0.0978	0.0485
TGCF	0.0160	0.0945	0.0472

Explanation and Rationale: Echoing 'Reliability' and 'Usability' aspects of ISO/IEC 25010:2011 [35], these metrics measure the breadth of product recommendations and their relevance, ensuring the system covers a wide spectrum of user preferences.

4) Popularity Bias and User Engagement

Metrics: Popularity Score

$$\text{Popularity Bias Score} = \text{Popularity Score} \tag{4.9}$$

Explanation and Rationale: Reflective of the 'Quality in Use' aspect [35], the Popularity Score metric is chosen to evaluate user engagement and satisfaction, focusing on how well the system balances popular and personalized recommendations.

4.5.3 Overall Performance Score

The overall performance of a CFRS is calculated as an aggregate of the individual dimensions:

$$\begin{aligned}\text{Overall Performance} = \frac{1}{4}(&\text{Accuracy and Predictive Performance} \\ &+ \text{Ranking Quality} \\ &+ \text{Diversity and Reliability} \\ &+ \text{Popularity Bias and User Engagement})\end{aligned} \tag{4.10}$$

4.5.4 Model Evaluation Results

The evaluation scores of various recommender models, based on this simplified methodology and in alignment with ISO/IEC 25010:2011 standards, are presented in Table 4.9:

Patterns and Themes

- **High Performers:** On the other hand, TGCF performs well in terms of the perfect scores in three important quality metrics: accuracy, ranking quality, and user engagement. In general, this model gets the maximum performance score among all models, which means it is preferable to use in most of the recommendation scenarios.

- **Specialized Performance:** Aside from TGCF, SGL attains the maximum Overall Performance, mean-

Table 4.9: Evaluation Scores for Different Models

Model	Accuracy and Predictive Performance	Ranking Quality	Diversity and Reliability	Popularity Bias and User Engagement	Overall Performance
DiffRec	4.996	4.272	2.574	3.947	3.947
ENMF	4.996	4.194	2.795	4.158	4.036
HMLET	4.998	4.334	2.796	4.211	4.085
LDiffRec	1.539	1.136	2.820	1.158	1.663
LightGCN	4.995	4.395	2.894	4.368	4.163
RecVAE	4.888	4.510	2.799	3.947	4.036
SGL	4.993	4.239	2.958	4.737	4.232
SimGCL	4.998	4.257	2.574	3.737	3.891
SimpleX	4.926	3.787	2.979	4.105	3.949
SpectralCF	1.000	1.000	3.000	1.000	1.500
XSimGCL	4.995	4.515	2.713	3.895	4.029
TGCF	5.000	5.000	3.373	5.000	4.593

ing it has more balanced performance than the other models, with exception to TGCF.

- **Low Performing Models:** LDiffRec and SpectralCF yield weak predictive power for most of the considered metrics. This means that their scores are considerably lower for the first two metrics, which include accuracy and predictive performance. This clearly indicates the limitation of effective user preference prediction and item ranking.

- **Variability in Popularity Bias and Engagement:** What is interesting is the variability of the models when handling popularity bias and the way they can engage users—from low scores for SpectralCF and LDiffRec to higher scores for TGCF. That is, the particular model to use in achieving goals regarding user engagement with a platform may vary.

Implications for Model Selection

Implications for the Model Selection The provided evaluation scores hint at where the strengths are in the recommendation model, and where its weaknesses are. In other words, a model like TGCF should be selected when focusing on having the highest accuracy regarding a user's engagement. Or in other words, a model like SGL may be more appropriate for platforms that emphasize diversity and content strategies. Proper selection among these models is based on comprehensive evaluation metrics that can lead to huge effectiveness in a recommendation system, affecting user satisfaction, engagement with the platform, and hence overall content discoverability.

4.6 Conclusions

The use of the ISO/IEC 25010:2011 standard-based framework for the evaluation of CFRSs in the adoption of the strategic approach is done effectively to handle the problems in the problem statement. The framework ensures that the users of the RS get better, more relevant, and more accurate recommendations, including novel and diverse ones. It also encourages adding scope, that is, a more comprehensive collection of products, including long-tail ones, by using metrics like Shannon Entropy and Item Coverage. This increases the user's exploration and solves the sparsity problem.

This nature of the framework, which is concerned with fairness, will be good in identifying and correcting biases in favor of popular items or those towards any particular demographic group, in the interest of a diverse representation of user interests. It helps to address the cold start issue by evaluating the capability of systems in handling new users or items with very limited interaction data.

Novel design in the TGCF model, which includes temporal characteristics, makes it possible to carry out inferences for more accurate and context-aware recommendations, like item popularity and diversity metrics. Exactly the time that the TGCF captures the temporal dynamics of user interest, there is an adaptable model that further improves the whole user experience of timely and reflective of the current trend recommendations.

Also, the challenge of reproducibility and the danger of overfitting are addressed by the comprehensive quality attributes of the ISO/IEC 25010:2011 standard. It is ensured through system testing to prove its robustness within many scenarios; therefore, it is not only effective by nature but proven under practical conditions. The framework also ensures the identification and correction of any imbalance or systematic disharmony of the systems through an evaluation of the system across many quality dimensions, some of which include reliability and usability. These guidelines make possible a structured review that leads to further improvements in system designs and functionalities, making CFRSs strong, fair, and adaptively relevant to the different needs of the users. Thus, the integration of this framework within the development and improvement of CFRSs enhances not only the technical capability of such systems but also secures the delivery of excellent and equal user experiences.

Chapter 5

Solution 2 - A Hybrid Graph Convolutional Networks for Optimizing Ranking and Long-tail Awareness of E-commerce Recommendation Systems

5.1 Introduction

It is worth noting that this chapter is based on a paper accepted with a major revision in an academic journal[1]. In this chapter, we present a CFRS that mitigates the challenges discussed in section 1.4, which is described in more detail in Table 1(in Appendix .1). We describe a general CFRS framework that gives new insight to the issue of improving diversity and ranking of recommended items in E-commerce. Meanwhile, the proposed model's effectiveness is validated through experiments over three benchmark datasets and compared with five state-of-the-art CFRS models. The research findings are also tested for statistical significance to support the belief that performance differences among various CFRS models exist.

5.1.1 Preliminary

Throughout this section, we define some of the important concepts for this study[2]. The definitions are as follows:

Bipartite Graph. We consider the simple graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ as a bipartite graph if the vertex set $\mathcal{V}$ can be partitioned into two disjoint subsets of users and items, $\mathcal{U} \cup \mathcal{I}$, such that every edge $e \in \mathcal{E}$ has the form $e = (u, i)$ where $u \in \mathcal{U}$ and $i \in \mathcal{I}$. Accordingly, we define the bipartite graph as $\mathcal{B} = \{\mathcal{U}, \mathcal{I}, \mathcal{E}\}$.

Adjacency Matrix. The bipartite graph $\mathcal{B}$ has an associated adjacency matrix $\boldsymbol{A}$, which can be defined as:

$$\boldsymbol{A}_{N\times N} = \begin{bmatrix} 0 & \boldsymbol{R} \\ \boldsymbol{R}^{\intercal} & 0 \end{bmatrix}, \tag{5.1}$$

where $N \times N$ are dimensions for $\boldsymbol{A}$, $\boldsymbol{R}$ is an implicit feedback matrix with dimensions $|\mathcal{U}| \times |\mathcal{I}|$, $\boldsymbol{R}_{u,i} = 1$ if there is a relation between $u \in \mathcal{U}$ and $i \in \mathcal{I}$, otherwise $\boldsymbol{R}_{u,i} = 0$.

Normalized Adjacency and Laplacian Matrices

[1]This is the first accepted paper with revision in Section 1.8

[2]Extended info can be found in Section 2.1.4

we define the normalized adjacency matrix as

$$\boldsymbol{A}^{\eta} \equiv \boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}, \tag{5.2}$$

where $\boldsymbol{A}$ is the adjacency matrix of $\mathcal{G}$ and $\boldsymbol{D} = diag(d)$ for $d(i)$ the degree of node i.

For a graph $\mathcal{G}$ (with no isolated vertices), we can see that

$$\boldsymbol{D}^{-1/2} = \begin{bmatrix} \frac{1}{\sqrt{d(1)}} & \dots & \dots & 0 \\ 0 & \frac{1}{\sqrt{d(2)}} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{1}{\sqrt{d(n)}} \end{bmatrix} \tag{5.3}$$

The un-normalized Laplacian matrix [147] is defined as:

$$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A} \tag{5.4}$$

where $\boldsymbol{L}$ has a set of eigenvectors $\boldsymbol{U} = (\mathrm{u}_1, \mathrm{u}_2, .., \mathrm{u}_N)$ and accompanied with set of eigenvalues $\Lambda = \{\lambda_l\}_{l=1}^{N}$, which are real and non-negative. Thus, there are variants of the graph Laplacian matrix, each with distinct properties. For instance, the normalized Laplacian matrix is a symmetric normalized version of $\boldsymbol{L}$, defined as:

$$\boldsymbol{L}^{\eta} = \boldsymbol{I} - \boldsymbol{A}^{\eta} = \boldsymbol{I} - \boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2} \tag{5.5}$$

On the other hand, $\boldsymbol{L}^{RW} = \boldsymbol{D}^{-1}\boldsymbol{L}$ is a random walk graph Laplacian, which is an asymmetric normalized version of $\boldsymbol{L}$.

Based on Equation 5.5, $\boldsymbol{L}^{\eta}$ is symmetric, square, and normalized matrix. Consequently, $\boldsymbol{L}^{\eta}$ can be eigen-decomposed as follows:

$$\boldsymbol{L}^{\eta} = \boldsymbol{U}\Lambda\boldsymbol{U}^{\boldsymbol{T}} \tag{5.6}$$

where $\boldsymbol{L}^{\eta}$ has a set of orthogonal eigenvectors $\boldsymbol{U} = (\mathrm{u}_1, \mathrm{u}_2, .., \mathrm{u}_N)$ and accompanied with set of eigenvalues $\Lambda = diag([\lambda_1, \lambda_2, \dots, \lambda_N])$. Furthermore, the orthogonal square matrix $\boldsymbol{U}$ can be expressed as follows:

$$\boldsymbol{I} = \boldsymbol{U}\boldsymbol{U}^{\boldsymbol{T}} = \boldsymbol{U}^{\boldsymbol{T}}\boldsymbol{U} \tag{5.7}$$

where $\boldsymbol{I}$ is the identity matrix.

In next section we introduce the research problem.

5.1.2 The problem and implications.

Figure 5.1 demonstrates the implications of problems in RSs accuracy. Consequently, in this research context, we attempt to utilize spectral-based CF modeling to include more Long-tail items, knowing that previous spectral-based CF models are undervalued in comparison with spatial-based CF models [6, 7, 61]. Given a bipartite graph $G = (\mathcal{U}, \mathcal{I}, E)$, where $\mathcal{U}$ is the set of users, $\mathcal{I}$ is the set of items, and E is the set of edges representing the interactions between users and items, we want to learn a function $f : \mathcal{U} \times \mathcal{I} \rightarrow \mathbb{R}$ that predicts the rating that a user u would give to an item i.

For each user $u \in \mathcal{U}$, let $\mathcal{I}_u^+ \subseteq \mathcal{I}$ be the set of popular items that the user has interacted with, and $\mathcal{I}_u^- \subseteq \mathcal{I}$ be the set of long-tail items that the user has not interacted with.

As described in Table 1(in Appendix .1) and the challenges in Section 1.4, the objective is to expand the diversity of the recommended list of items from the list of long-tail items $\mathcal{I}_u^-$, and arrange them in descending order, where items with highest predicted ratings are at the top of the list. This can be formulated as finding a

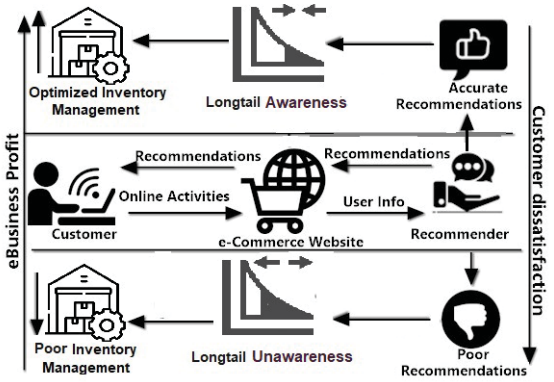


Figure 5.1: The motivation behind the research.

list of items $I_u^* \subseteq \mathcal{I}_u^-$, sorted in descending order by the predicted ratings $f(u,i)$, where $i \in I_u^*$.

SpectralCF uses graph Fourier transform (GFT)[3] to transform the bipartite graph signals into the spectral domain, which requires eigen-decomposition. LSGCN proposes a new design based on integrating simple GCN design with propagation control to improve learning from the domains of user-item bipartite graphs and make rank-aware and longtail-aware recommendations. It transforms the spectral learning problem into a semi-spatial learning problem, with competitive performance compared to state-of-the-art spatial-based CF models. LSGCN is expected to make the following contributions:

- In contrast to SpectralCF, LSGCN proposes a novel design based on integrating simple GCN with propagation control to improve learning from the domains of user-item bipartite graphs and make Rank-aware and Longtail-aware recommendations.
- Unlike traditional spectral collaborative filtering models [6,7,61], LSGCN has very competitive performance with state-of-the-art spatial-based CF models.
- Instead of mitigating the spectral learning problem through improving eigen-decomposition and GFT, LSGCN transforms the spectral learning problem into a semi-spatial learning problem.
- Eventually, proposing an optimized spectral-based CF model will bring more attention to this undervalued area and contribute positively to the research efforts for overcoming the challenges addressed in Figure 5.1.

5.2 Method

The following sub-sections discuss the proposed model:

5.2.1 Approximating GFT .

Referring to Equations 5.5 and 5.6, literature shows that the normalized graph laplacian might be regarded as an approximation to $\boldsymbol{U}\Lambda\boldsymbol{U}^{\boldsymbol{T}}$ [49]. Kipf and Welling [148] proposed that the normalized graph Laplacian is an effective approximation of the graph Fourier transform in capturing local graph structure. Similarly, Defferrard et al. [149] employed the normalised graph Laplacian as an approximation of the graph Fourier transform and discovered that it is correct for graphs with smooth spectral density and performs well in applications such as semi-supervised classification and link prediction. Similarly to Chebyshev Polynomials, the accuracy of $\boldsymbol{I} - \boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}$ as an estimate of $\boldsymbol{U}\Lambda\boldsymbol{U}^{\boldsymbol{T}}$, on the other hand, is dependent on the features of the graph and the particular application [49].

[3]Further information about GFT is available at Section 2.1.6

At the same time, employing a diagonal learnable parameter matrix with GCNs can enhance approximation accuracy. The learnable parameter matrix may change the weights allocated to each node in the graph, improving approximation accuracy for various graph shapes and applications. Several research studies have employed diagonal learnable parameter matrices with GCNs to increase graph Fourier transform accuracy, Xu et al. [150] proposed a multi-scale GCN with diagonal learnable parameter matrices for semi-supervised node classification tasks. Referring to Equations 5.5 and 5.6, we conduct an experiment to validate the claim that $\boldsymbol{I} - \boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}\boldsymbol{x}$ can efficiently approximate $\boldsymbol{U}\Lambda\boldsymbol{U}^{\boldsymbol{T}}\boldsymbol{x}$. Due to the significance of this claim to our proposed methodology, we validate it before the experiments section in a structured manner, as follows:

Experiment Overview

Equation 5.8 aims to test if the Normalized Laplacian Transform(NLT), $I - D^{-1/2}AD^{-1/2}\boldsymbol{x}$ can approximate the basic GFT, $\boldsymbol{U}\Lambda\boldsymbol{U}^{\boldsymbol{T}}\boldsymbol{x}$. Where $\boldsymbol{x} \in \mathbb{R}^{|\mathcal{N}|\times 1}$. We represent the approximation as follows:

$$\boldsymbol{I} - \boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}\boldsymbol{x} \approx \boldsymbol{U}\Lambda\boldsymbol{U}^{\boldsymbol{T}}\boldsymbol{x} \tag{5.8}$$

We designate the null hypothesis (H_0) and the alternative hypothesis (H_1) as follows:

$$\begin{aligned} &H_0 : \mu_1 = \mu_2 \\ &H_1 : \text{the two group means are not equal} \end{aligned} \tag{5.9}$$

where μ_1, μ_2 denote the population means for RMSE and MAE. The experiment aimed to compare the performance of two methods for predicting user-item preferences in bipartite graphs. We conducted the experiment on 100 bipartite graphs, each with between 500 and 5000 nodes.

Methodology

For each graph, we generated a random user-item matrix and computed the ground truth by matrix multiplication with the bipartite adjacency matrix. We then used both methods to make predictions and computed the root mean squared error (RMSE) and mean absolute error (MAE) for each method. Finally, we conducted t-tests and Wilcoxon signed-rank tests to compare the predictions of the two methods.

Results

Table 5.1 shows the mean RMSE and MAE values for each method, along with the p-values for the t-tests and Wilcoxon signed-rank tests, both uncorrected and Bonferroni-corrected.

Table 5.1: Comparison of NLT and GFT for predicting user-Item preferences in Bipartite graphs, results and statistical analysis

Method	Mean RMSE		Mean MAE		P-value	
	NLT	GFT	NLT	GFT	t-test	Wilcoxon signed-rank test
Value	0.7271	0.7272	0.5643	0.5645	0.7038	0.8025
Corrected	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000

Note that GFT is slightly better than NLT in terms of mean RMSE and MAE, but the differences are not statistically significant. In Experiment 2, the t-test showed a p-value of 0.7038 and the Wilcoxon signed-rank test showed a p-value of 0.8025, both above the significance level of 0.05. After Bonferroni correction, all p-values are above the corrected significance level of 0.025, indicating that there is no statistically significant difference between the two methods.

A corrected p-value of 1.0000 means that after Bonferroni correction, the p-value is still greater than the significance level (i.e., 0.05). In other words, the difference between the two methods being compared is not statistically significant at the 0.05 level of significance.

5.2.2 Simplified Spectral Collaborative Filtering.

We begin by approximating the convolution filters through using first z polynomials as follows:

$$g_{\varphi}(\Lambda) \approx \sum_{z=0}^{Z} \varphi_z \Lambda^z \tag{5.10}$$

The learning complexity in the filter is $O(Z)$, where Z is a hyper-parameter. To avoid over-fitting, we chose the order of the polynomial $Z = 1$. By substituting Equation 5.10 into Equation 2.5, we have:

$$\begin{bmatrix} \boldsymbol{x}_{\eta}^{u} \\ \boldsymbol{x}_{\eta}^{i} \end{bmatrix} = (\varphi_0 U U^{\mathrm{T}} + \varphi_1 U \Lambda U^{\mathrm{T}}) \begin{bmatrix} \boldsymbol{x}^{u} \\ \boldsymbol{x}^{i} \end{bmatrix} \tag{5.11}$$

where the learned graph signals are represented as vectors $\boldsymbol{x}_{\eta}^{u} \in \mathbb{R}^{|u| \times 1}$ and $\boldsymbol{x}_{\eta}^{i} \in \mathbb{R}^{|i| \times 1}$.

Based on previous research approaches [148], we simplify the spectral learning problem by substituting both Equations 6.2 and 5.7 in Equation 5.11 to get the following equation:

$$\begin{bmatrix} \boldsymbol{x}_{\eta}^{u} \\ \boldsymbol{x}_{\eta}^{i} \end{bmatrix} = (\varphi_0 \boldsymbol{I} + \varphi_1 \boldsymbol{L}^{\eta}) \begin{bmatrix} \boldsymbol{x}^{u} \\ \boldsymbol{x}^{i} \end{bmatrix} \tag{5.12}$$

we utilize Equation 5.12 to avoid the computational challenges in eigen-decomposition and GFT. Furthermore, we further simplify Equation 5.12 by decreasing the number of parameters through setting $\gamma = \varphi_0 = \varphi_1$, as follows:

$$\begin{bmatrix} \boldsymbol{x}_{\eta}^{u} \\ \boldsymbol{x}_{\eta}^{i} \end{bmatrix} = \gamma(\boldsymbol{I} + \boldsymbol{L}^{\eta}) \begin{bmatrix} \boldsymbol{x}^{u} \\ \boldsymbol{x}^{i} \end{bmatrix} \tag{5.13}$$

Next, we further simplify the equation by substituting Equation 5.5 in Equation 5.13, as follows:

$$\begin{bmatrix} \boldsymbol{x}_{\eta}^{u} \\ \boldsymbol{x}_{\eta}^{i} \end{bmatrix} = \gamma(\boldsymbol{I} + \boldsymbol{I} - \boldsymbol{A}^{\eta}) \begin{bmatrix} \boldsymbol{x}^{u} \\ \boldsymbol{x}^{i} \end{bmatrix} \tag{5.14}$$

Moreover, we further simplify Equation 5.14 as follows:

$$\begin{bmatrix} \boldsymbol{x}_{\eta}^{u} \\ \boldsymbol{x}_{\eta}^{i} \end{bmatrix} = \gamma(2\boldsymbol{I} - \boldsymbol{A}^{\eta}) \begin{bmatrix} \boldsymbol{x}^{u} \\ \boldsymbol{x}^{i} \end{bmatrix} \tag{5.15}$$

We generalize both graph signals $\boldsymbol{x}^{u} \in \mathbb{R}^{|u| \times 1}$ and $\boldsymbol{x}^{i} \in \mathbb{R}^{|i| \times 1}$ to signal embedding matrices $\boldsymbol{X}^{u} \in \mathbb{R}^{|u| \times d}$ and $\boldsymbol{X}^{i} \in \mathbb{R}^{|i| \times d}$, respectively. Based on previous research approaches [28, 148, 151], we generalize the filter parameter γ to a trainable parameters vector $\boldsymbol{\gamma}^{v} \in \mathbb{R}^{(N) \times 1}$. Each parameter $\boldsymbol{\gamma}_{i}^{v}$ learns to control the propagation process for each node i, transforming LSGCN to an adaptive GCN. Then, we use sigmoid activation to limit the weight values between 0 and 1. To illustrate, we construct a new trainable parameter matrix as the diagonal matrix $\boldsymbol{\Gamma} = diag(\sigma(\boldsymbol{\gamma}^{v}))$, where $\boldsymbol{\Gamma} \in \mathbb{R}^{N \times N}$.

As a result, our proposed spectral convolution operation is changed as following:

$$\begin{bmatrix} \boldsymbol{X}_{\eta}^{u} \\ \boldsymbol{X}_{\eta}^{i} \end{bmatrix} = \Gamma(2\boldsymbol{I} - \boldsymbol{A}^{\eta}) \begin{bmatrix} \boldsymbol{X}^{u} \\ \boldsymbol{X}^{i} \end{bmatrix} \tag{5.16}$$

Equation 5.16 shows the simplified convolution operation where the convolution process is parameterized by $\boldsymbol{A}^{\eta}$ and Γ. Conversely, most previous spectral collaborative filtering models require U and Λ as essential parameters to convolution operations, which complicates the convolution process and requires significantly higher resources [6, 7, 61].

In the following section, we present our proposed graph neural network architecture.

5.2.3 The Graph Convolutional Network Architecture

Integrating the convolution operation in Equation 5.16 with a GCN leads to the following l-layer-based structure:

$$\begin{bmatrix} \boldsymbol{X}_l^u \\ \boldsymbol{X}_l^i \end{bmatrix} = \Gamma_{l-1}(2\boldsymbol{I} - \boldsymbol{A}^\eta) \begin{bmatrix} \boldsymbol{X}_{l-1}^u \\ \boldsymbol{X}_{l-1}^i \end{bmatrix},\ l = 1..L \tag{5.17}$$

Noticeably, we don't apply non-linear activation to the outcome of the GCN, which is recommended by well-known research [23]. $\boldsymbol{\Gamma}_l \in \mathbb{R}^{N \times N}$ is the filter parameters matrix for the l_{th} layer. $\begin{bmatrix} \boldsymbol{X}_0^u \\ \boldsymbol{X}_0^i \end{bmatrix}$ and $\boldsymbol{\Gamma}_0$ are initialized randomly and used as input for first l_{th} layer. Both $\boldsymbol{X}_l^u \in \mathbb{R}^{|\mathcal{U}| \times d}$ and $\boldsymbol{X}_l^i \in \mathbb{R}^{|\mathcal{I}| \times d}$ present the learned convoluted graph signals in the current l_{th} layer while d presents the embedding size. The results of Equation 5.17 is concatenated into latent factors of users and items, as follows:

$$\begin{bmatrix} \boldsymbol{W}^u \\ \boldsymbol{W}^i \end{bmatrix} = \left[\begin{bmatrix} \boldsymbol{X}_0^u \\ \boldsymbol{X}_0^i \end{bmatrix} \mid \ldots\ldots \mid \begin{bmatrix} \boldsymbol{X}_L^u \\ \boldsymbol{X}_L^i \end{bmatrix} \right], \tag{5.18}$$

Both matrices $\begin{bmatrix} \boldsymbol{W}^u \\ \boldsymbol{W}^i \end{bmatrix}$ are LSGCN's latent matrices of users and items, where $\boldsymbol{W}^u \in \mathbb{R}^{|\mathcal{U}| \times Ld}$ and $\boldsymbol{W}^i \in \mathbb{R}^{|\mathcal{I}| \times Ld}$. We represent the final embedding matrix as $\boldsymbol{E} = \begin{bmatrix} \boldsymbol{W}^u \\ \boldsymbol{W}^i \end{bmatrix}$, where $\boldsymbol{E} \in \mathbb{R}^{|\mathcal{N}| \times Ld}$.

5.2.4 Optimization

Concerning the loss function, we select a pairwise loss approach with Bayesian Personalized Ranking (BPR). The following presents the BPR equation for our proposed model:

$$Loss = \sum_{(o,c,c') \in \mathrm{D}} - \ln \sigma(\boldsymbol{w}_o^{u\intercal} \boldsymbol{w}_c^i - \boldsymbol{w}_o^{u\intercal} \boldsymbol{w}_{c'}^i) + \alpha \| \boldsymbol{E} \|_2^2 \tag{5.19}$$

where $D = \{(o, c, c') | (o, c) \in \mathcal{R}^+,\ (o, c') \in \mathcal{R}^-\}$ represents BPR training data, $\mathcal{R}^+$ represent the detected interactions that users' liked, clicked, or viewed items. Meanwhile, $\mathcal{R}^-$ represent the remaining undetected interactions, $\boldsymbol{E}$ represents the final embedding matrix, and α parameter prevents over-fitting through adjusting the regularization term. We utilize Adam optimizer to optimize the prediction model and update the model parameters. Algorithm 1 represents further description to LSGCN algorithm. At the same time, Figure 5.2 represents LSGCN as a flowchart.

In the next section we conduct the experiments to validate our assumptions about the proposed model.

5.3 Experiments

In this section we conduct seven experiments to evaluate the performance of LSGCN against five CFRS models. The research experiments should answer the following four chapter questions:

CHQ1. How does LSGCN compare to the other CFRS models in terms of rank-awareness and Longtail-awareness of the recommended list of items?

CHQ2. Is LSGCN model overfitted to specific hyper-parameter settings, such as the number of epochs and the top k items

CHQ3. How does sparsity problem in CFRSs impact LSGCN recommendations?

CHQ4. Are the differences in performance between LSGCN and the other CFRS models significant?

The following sub-sections describe the datasets, evaluation metrics, implementation details, and baseline

Algorithm 2 : A Simplified Pseudo-code Version of LSGCN.

Input: Training set $D = \{(o, c, c')|(o, c) \in \mathcal{R}^+, \ (o, c') \in \mathcal{R}^-\}$, Batch Size *S*, Embedding Size *M* , Number of Epochs *E*, Number of Layers *L*.

Output: parameter set $\Phi = \{\boldsymbol{\gamma}_0^v, \boldsymbol{\gamma}_1^v, \ldots, \boldsymbol{\gamma}_{l-1}^v, \boldsymbol{X}_0^u, \boldsymbol{X}_0^i\}$.

Initialize $\boldsymbol{X}_0^u$,$\boldsymbol{X}_0^i$, $\boldsymbol{\gamma}_0^v$ using Glorot normal initializer [152].

for $e = 1$ to E **do**

B_e=GenerateBatch(*D*,*S*)

for $l = 1$ to L **do**

Calculate $\boldsymbol{\Gamma}_{l-1} = diag(\sigma(\boldsymbol{\gamma}_{l-1}^v))$

Calculate $\begin{bmatrix} \boldsymbol{X}_\eta^u \\ \boldsymbol{X}_\eta^i \end{bmatrix} = \Gamma(2\boldsymbol{I} - \boldsymbol{A}^\eta) \begin{bmatrix} \boldsymbol{X}^u \\ \boldsymbol{X}^i \end{bmatrix}$

Concat $\begin{bmatrix} \boldsymbol{X}_l^i \\ \boldsymbol{X}_l^u \end{bmatrix}$ into $\begin{bmatrix} \boldsymbol{W}^i \\ \boldsymbol{W}^u \end{bmatrix}$

end for

Calculate gradients for Φ_e .

Update Φ_{e+1} using Adam optimizer [153].

end for

return parameters set Φ_E

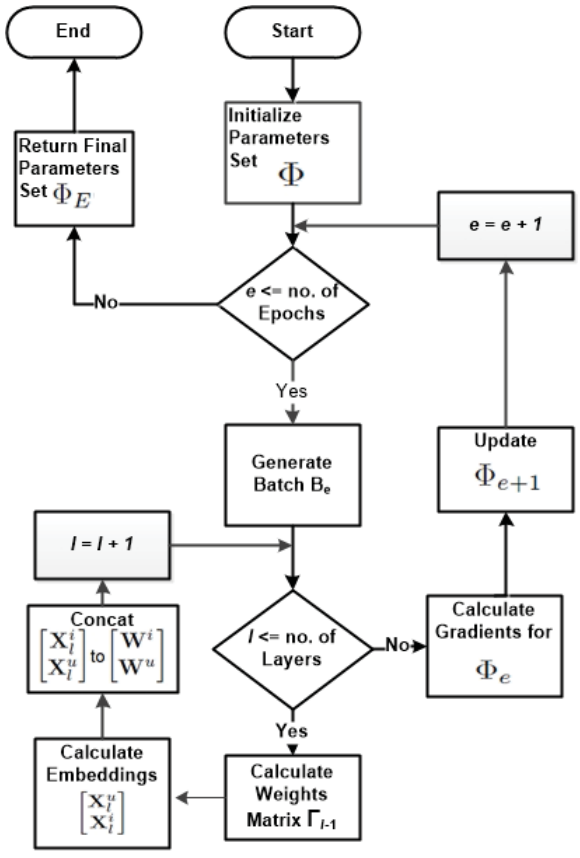


Figure 5.2: A flowchart represnting LSGN model.

methods. For all experiments in this section, we include the results for some datasets as the remaining results follow the same trend.

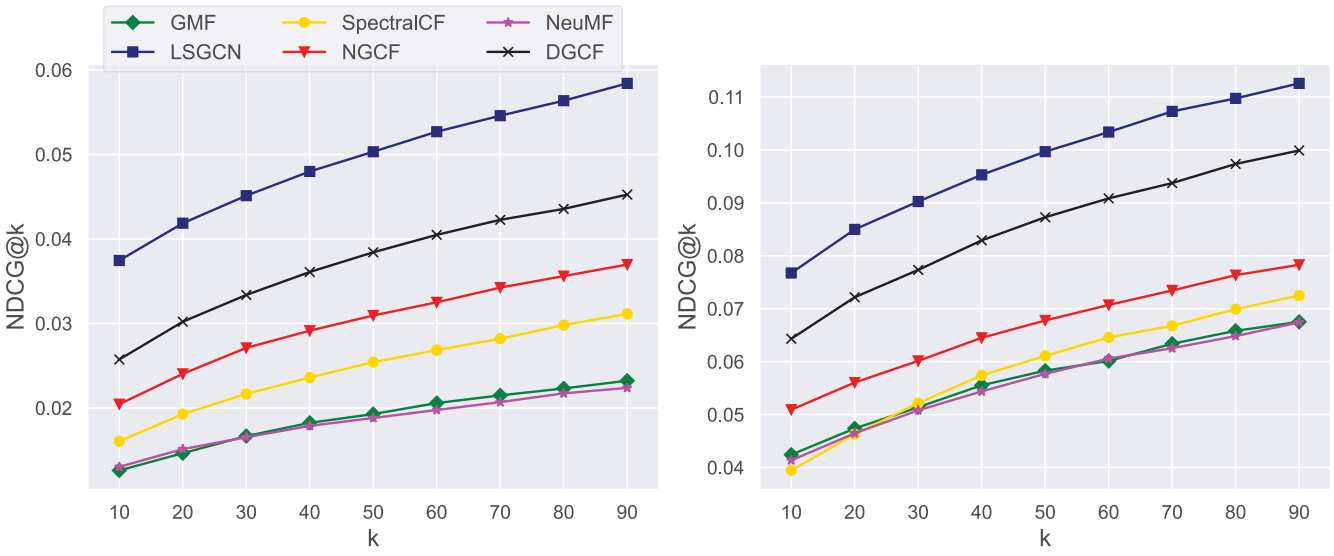
Implementation details. To improve our experiments, we followed RecBole's guidelines and recommendations [143–146]. RecBole is an end-to-end comprehensive and modular framework engineered for developing and experimenting with recommendation algorithms.

For this chapter context[4], we utilized the Gowalla dataset [61], Amazon Musical Instruments dataset(MI) [154], and Amazon Arts, Crafts and Sewing dataset(ACS) [154], we excluded the information about users and items that have less than twenty interactions for the Gowalla dataset. Furthermore, we excluded information with fewer than 10 interactions for the Amazon Musical Instruments dataset, and with fewer than twenty interactions for the Amazon Arts, Crafts, and Sewing dataset. This stage of data preprocessing allowed us to improve the quality of our research by considering only sufficiently numerous and significant user interactions.

The dataset is split into training and testing sets. The training set consists of 80% of the data, and the testing set consists of the other 20%. We interpret each event in the training set as a positive interaction and randomly choose one negative interaction from the training set. Furthermore, we optimized the hyper-parameters through grid-search. We set the embedding size in range $\{64, 128\}$ for both LSGCN and SpectralCF, and we optimized the learning rate using grid search in set $\{0, 0.001, 0.01, \ldots, .1\}$. We optimized LSGCN with Adam optimizer [153] while we set the batch size to 1024. We also set the depth of LSGCN neural network into three layers while we initialized the model parameters with the Glorot normal initializer [152]. Due to the space limitation, we choose to include the results for some research results, knowing that the remaining results follow similar trend.

Information about the used baselines are available at Section 2.1.12 whereas information about used metrics are available at Section 2.1.11. In the following sub-sections we present the research experiments:

5.3.1 The effect of varying the value of top k on the performance measurements of the CFRS models(CHQ2)



(a) MI dataset

(b) ACS dataset

Figure 5.3: Performance comparison of LSGCN and baselines on the ACS and MI dataset in terms of NDCG@k.

[4]Further information about the Datasets is available at Section 2.1.12

Both figures 5.3a and 5.3b compare the impact of varying the value of variable k on the performance of the CFRS models.

We have the following observations about the results:

- Both GMF and NeuMF show very similar performance on NDCG@k, reflecting less rank-awareness and less sensitivity to increasing top-k value. Furthermore, both GMF and NeuMF don't directly model connection throughout the embedding learning process, which may result in suboptimal representations.
- When compared to both GMF and NeuMF, SpectralCF shows higher sensitivity for increasing top-k value, particularly on MI dataset. Thus, SpectralCF performance is still below NGCF, DGCF, and LSGCN.
- LSGCN supercedes the remaining models.

Table 1(in Appendix .1) shows the experiment's approach to handling the challenges described in Section 1.4.

5.3.2 Investigating the impact of varying top k items for Coverage@k measurement over three CFRS models(CHQ1 & CHQ4)

Table 5.2: Performance comparison of LSGCN and baselines with increasing top k items for Coverage@k performance measurement. Boldfaced numbers indicate best performance over all models.

Dataset	Method	Coverage%									
		k=10	k=20	k=30	k=40	k=50	k=60	k=70	k=80	k=90	k=100
ACS	SpectralCF	17.07	24.24	29	32.6	35.5	37.9	40	41.83	43.51	45.03
	DGCF	26.48	38.43	46.30	51.93	56	59	61.28	63.09	64.59	65.84
	LSGCN	**52.71**	**68.91**	**75.82**	**78.98**	**80.76**	**81.85**	**82.6**	**83.07**	**83.41**	**83.65**
MI	SpectralCF	9.44	13.33	16	18	19.77	21.22	22.5	23.6	24.61	25.55
	DGCF	22.97	33.81	41.90	48.13	53	56.75	59.62	61.78	63.51	64.91
	LSGCN	**41**	**57.6**	**67.31**	**73.33**	**77.03**	**79.34**	**80.77**	**81.7**	**82.33**	**82.76**
Gowalla.	SpectralCF	29.56	39.38	46.08	51.15	54.83	57.94	60.63	62.93	64.85	66.64
	DGCF	32.60	46.92	57.21	64.66	70.36	74.68	78.22	81.23	83.68	85.76
	LSGCN	**50**	**65.48**	**74.64**	**80.76**	**85.06**	**88.06**	**90.45**	**92.17**	**93.47**	**94.56**

The coverage of a RS is concerned with the domain of items that the RS is capable of recommending. For instance, most RSs prefer popular items over Long-tail or cold-start items. As Figure 5.1 shows, RSs with limited item coverage confines the user's recommendations, which complicates inventory management , affects user satisfaction, and decrease total system sales. Looking at table 5.2, we have the following observations:

- In general, we can see that as k (the number of recommendations per user) grows, so does coverage. This is to be anticipated, as recommending more products to each user means that more items are being recommended in general, and therefore a bigger set of items has a likelihood of being recommended to at least one user.
- The difference in coverage of LSGCN from the baselines is higher for lower k values, especially when k is less than 50. Therefore, having a higher item coverage@k on lower k values implies that the algorithm is capable of providing a broader range of recommendations to new users in cold-start situations.

- The higher item coverage@k on lower k values also suggests that the algorithm is more efficient in recommending long tail items to users, even when considering a small number of top items .In other words, the algorithm is able to recommend a wider range of items, including those that are less popular or niche.

Table 1(in Appendix .1) shows the experiment's approach to handling the challenges described in Section 1.4.

5.3.3 Investigating the Impact of Varying Top k Items on Gini Index Across Three CFRS Models (CHQ1 & CHQ4)

The Gini Index in a CFRS quantifies the equity of item recommendation; lower values suggest a more equitable distribution across the spectrum of items, potentially counterbalancing the system's inherent preference for popular items over long-tail or cold-start items. By analyzing Table 5.3, the following observations can be made:

- **Decrease in Gini Index with increasing k:** Generally, as the number of top recommendations per user (k) increases, the Gini Index decreases. This trend is expected as expanding the recommendation list for each user naturally covers a broader range of items, hence improving the equity in item distribution.
- **Marked superiority of LSGCN:** Notably, LSGCN demonstrates a consistently lower Gini Index across all k values compared to baseline models SpectralCF and DGCF, especially evident at lower k values (e.g., $k = 10$). This suggests that LSGCN is particularly effective in recommending a diverse array of items early in the recommendation list, enhancing exposure to long-tail items.
- **Efficiency in long-tail recommendation:** The ability of LSGCN to maintain lower Gini Index values at lower k levels indicates its efficiency in promoting lesser-known or niche items. This capability is crucial for enhancing user discovery and satisfaction, providing a richer and more diverse user experience.

These insights underscore the importance of evaluating CFRSs not only for their accuracy but also for their fairness and inclusivity in item recommendation. The lower Gini Index values achieved by LSGCN highlight its potential in creating more balanced recommendation lists that cater to a wider range of user preferences and needs. Table 1(in Appendix .1) shows the experiment's approach to handling the challenges described in Section 1.4.

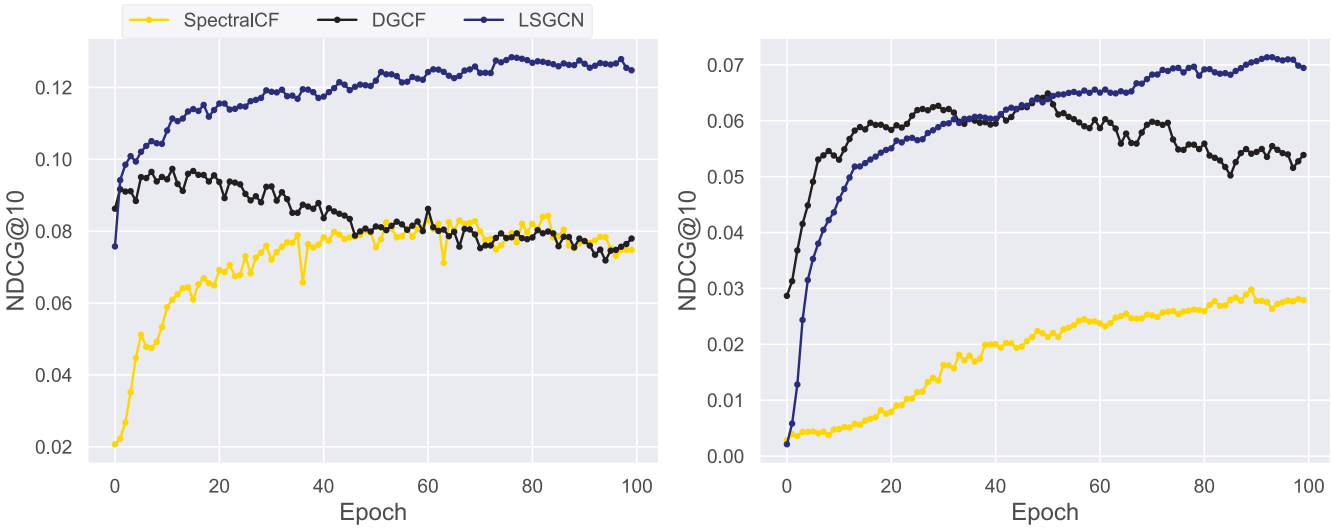
5.3.4 Evaluate the performance of three CFRS models with relation to the number of epochs (CHQ2).

Figure 5.4 compares Recall@20 for each epoch in LSGCN and SpectralCF. We have the following observations about figure 5.4 :

- LSGCN exhibits higher performance on both Gowalla and ACS datasets. Both datasets show that LSGCN learning improves steadily with increasing the number of epochs.
- DGCF performance is generally better than SpectralCF on both datasets, Gowalla and ACS. However, DGCF performance on Gowalla start to degrade after the twentieth epoch.
- LSGCN shows higher stability and improved learning with increasing the number of epochs. On the other hand, DGCF performance varies considerably according to the dataset. Table 1(in Appendix .1) shows the experiment's approach to handling the challenges described in Section 1.4.

Table 5.3: Performance comparison of LSGCN, DGCF, and SpectralCF with increasing top k items for Gini Index performance measurement on different datasets. Lower Gini Index values indicate better performance in terms of equity of item coverage.

Dataset	Method	Gini Index for k=									
		10	20	30	40	50	60	70	80	90	100
Arts	SpectralCF	0.962	0.949	0.940	0.933	0.927	0.922	0.918	0.914	0.910	0.907
	DGCF	0.929	0.905	0.892	0.885	0.880	0.877	0.875	0.873	0.872	0.871
	LSGCN	**0.739**	**0.689**	**0.671**	**0.664**	**0.660**	**0.659**	**0.657**	**0.656**	**0.656**	**0.655**
Gowalla	SpectralCF	0.930	0.907	0.889	0.876	0.863	0.853	0.843	0.834	0.826	0.819
	DGCF	0.910	0.868	0.834	0.806	0.781	0.759	0.739	0.720	0.703	0.687
	LSGCN	**0.846**	**0.794**	**0.757**	**0.728**	**0.703**	**0.682**	**0.663**	**0.646**	**0.631**	**0.618**
Musici	SpectralCF	0.983	0.978	0.974	0.971	0.969	0.966	0.964	0.963	0.961	0.960
	DGCF	0.928	0.906	0.893	0.884	0.879	0.874	0.872	0.870	0.869	0.868
	LSGCN	**0.804**	**0.752**	**0.725**	**0.709**	**0.699**	**0.693**	**0.689**	**0.686**	**0.684**	**0.682**



(a) Gowalla dataset.

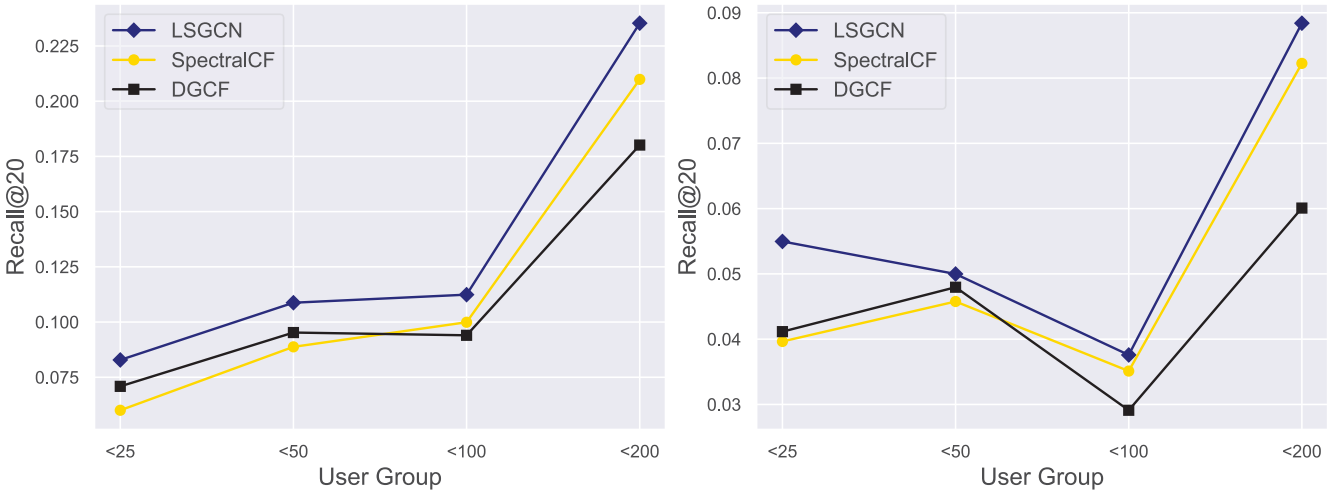
(b) ACS dataset.

Figure 5.4: Performance evaluation for the first 100 epochs in LSGCN, SpectralCF, and DGCF.

5.3.5 Comparing CFRS models with relation to different density groups (CHQ3).

Because few interactions of inactive users are inadequate to build high-quality representations, the sparsity issue frequently restricts the expressiveness of CFRSs. We examine if using connection information might assist to lessen this problem. We split Gowalla and MI dataset into four groups, where each group contains users who interacted with a specified range of items. To exemplify, we split users into four groups: (0, 25], (25, 50], (50, 100] and (100, 200]. Furthermore, we discard users who interacted more than 200 items. As figure 5.5 shows, Recall@20 correlates positively with the number of interactions for each group of users. We have the following observations about figure 5.5:

- Figure 5.5 shows that LSGCN supersedes all the other CFRS models, indicating better performance for either sparse or dense datasets.



(a) ACS Dataset

(b) MI Dataset

Figure 5.5: Comparing the performance of three CFRS models with varying density groups.

- LSGCN improved performance can be attributed to the simplified convolution process and the adaptive propagation control.
- Both figures 5.5a and 5.5b show that DGCF is better than SpectralCF in the first two groups.
- SpectralCF improved performance in the last two groups is not significant since the first two groups contain the vast majority of interactions between users and items.
- Figure 5.5 illustrates that the simplicity and propagation control in LSGCN considerably improves representation learning for inactive users since the collaboration signal can be captured efficiently. As a result, solving the sparsity problem in CFRSs may be promising with further research on LSGCN.

Table 1(in Appendix .1) shows the experiment's approach to handling the challenges described in Section 1.4.

5.3.6 Statistical Significance Analysis(CHQ4)

In this subsection, we need to determine if there a significant statistical difference in performance between LSGCN, DGCF and SpectralCF. Whereas ANOVA is founded on the premise that all sample populations are normally distributed, it is recognized to be resistant to minor deviations from this assumption. We perform one-way ANOVA for the accuracy scores data obtained by LSGCN, DGCF and SpectralCF on the ACS and Gowalla datasets. We designate the null hypothesis (H_0) and the alternative hypothesis (H_1) as follows:

$$\begin{aligned} H_0 &: \mu_1 = \mu_2 = \mu_3 \\ H_1 &: \text{not all group means are equal} \end{aligned} \tag{5.20}$$

where μ_1, μ_2, μ_3 denote the population means.

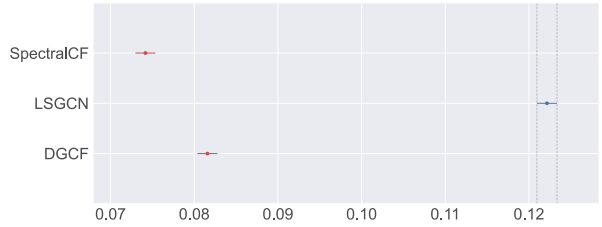
Looking at Table 5.4, these results correspond to the mean of 5 run times for the first 150 epochs of LSGCN, DGCF and SpectralCF, respectively. The p-values (< 0.05) show that the variations in accuracy means are statistically significant, where $\alpha = 0.05$ is the significance level. A significance level of 0.05 means that a 5% possibility of assuming that there is a difference when there is none.

Table 5.4: One-way ANOVA p-values for the accuracy scores data obtained by LSGCN, DGCF and SpectralCF on the Gowalla and ACS datasets.

Dataset	Measurement	p-value	Reject
Gowalla	NDCG@10	<0.0001	True
ACS	Recall@10	<0.0001	True

Group1	Group2	Adjusted p-value	Reject
DGCF	LSGCN	<0.0001	True
DGCF	SpectralCF	<0.0001	True
LSGCN	SpectralCF	<0.0001	True

(a) Analysis results for Gowalla dataset with NDCG@10.

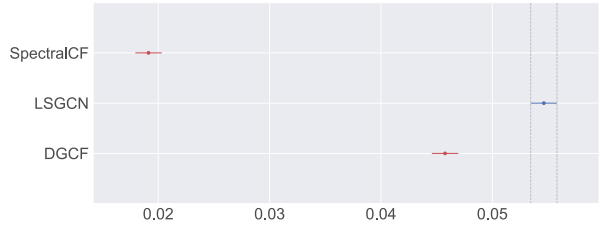


(b) Confidence Intervals Plot for Gowalla Dataset with NDCG@10.

Figure 5.6: Pairwise multiple comparison between LSGCN, DGCF and LightGCN methods using Tukey's test on the Gowalla dataset.

(a) Analysis results for ACS dataset with Recall@10.

Group1	Group2	Adjusted p-value	Reject
DGCF	LSGCN	<0.0001	True
DGCF	SpectralCF	<0.0001	True
LSGCN	SpectralCF	<0.0001	True



(b) Confidence Intervals Plot for ACS Dataset with Recall@10.

Figure 5.7: Pairwise multiple comparison between LSGCN, DGCF and LightGCN methods using Tukey's test on the ACS dataset.

So even though our ANOVA analysis demonstrates a considerable overall difference in group means, we need to know which groups have higher means by performing multiple pairwise comparison (post-hoc comparison) analysis using Tukey's test. Looking at table 5.6a and table 5.7a, for all experiments, the null hypothesis is rejected and we can conclude that the differences between different group means are statistically significant. However, we are not sure about the order of the three CFRS models in terms of performance.

In figure 5.6b and figure 5.7b, the 95% confidence intervals plots are displayed as horizontal bars. The blue bar shows the comparison interval for LSGCN mean while the red bars show DGCF and SpectralCF means. Clearly, there is no overlap between intervals. Furthermore, LSGCN shows the highest mean, followed by DGCF and SpectralCF, respectively. We omitted the performance that has a similar pattern due to space constraints. As expected, the results in Figure 5.6 and Figure 5.7 align with the results reported in the previous experiments. Table 1(in Appendix .1) shows the experiment's approach to handling the challenges described in Section 1.4.

5.4 Conclusions

In conclusion, this chapter introduces a novel recommendation model called LSGCN, which combines spectral and spatial methodologies in the field of recommendation systems. This model addresses the contradiction

present in SpectralCF, which is effective but suffers from complexity.

LSGCN simplifies the training process by removing the need for eigen-decomposition and nonlinear activation found in SpectralCF. By applying the TRIZ principle of 'Simplification,' we eliminate unnecessary complexity without compromising effectiveness.

Experimental results on real-world datasets demonstrate that LSGCN outperforms SpectralCF and other benchmark models in terms of recommendation accuracy. It performs well in cold-start scenarios and captures the long-tail distribution of items. This utilization of limitations as strengths aligns with the TRIZ principle of 'Inversion.'

LSGCN also achieves superior item coverage at lower values of k, indicating its ability to provide personalized and relevant recommendations to users. This contributes to user satisfaction and business outcomes, reflecting the TRIZ principle of 'Ideal Final Result.'

Statistical significance testing confirms the substantial and non-random differences between LSGCN and other CFRS models. By bridging spectral and spatial methods while simplifying computation and integrating graph convolution, LSGCN exemplifies the TRIZ principle of 'Integration.'

This study contributes to the advancement of spectral-based CFRS models for E-commerce recommendations and highlights the need for further research and development. With the guidance of TRIZ, we are inspired to innovate and overcome contradictions in our pursuit of optimizing recommendation systems.

Chapter 6

Solution 3 - Enhancing E-Commerce Recommendations with a Novel Scale-Aware Spectral Graph Wavelets Framework

6.1 Introduction

This chapter is based on a published research paper [2] about a CFRS called Scale-Aware Wave Graph Embedding (SAWE). We tailored SAWE to mitigate the challenges in section 1.4, further details available in Table 1(in Appendix .1).

SAWE is a CFRS specifically designed to augment the interpretation and learning from the spectral domains of bipartite graphs. This innovation seamlessly merges spectral theory and deep learning principles, tailor-made for the challenges of collaborative filtering. By taking a cluster-based approach to eigenvalues, SAWE applies a unique, adaptive smoothing function to each cluster, optimized using the Signal-to-Noise Ratio (SNR). This method allows SAWE to deliver a nuanced treatment of eigenvalues, ensuring a high-quality assessment and identification of pivotal information within clusters. Such an approach results in superior eigenvalue filtering, capturing intricate patterns across diverse scales. Moreover, the efficacy of SAWE has been empirically validated; it exhibits exceptional performance in addressing significant collaborative filtering challenges, such as enhancing recommendation diversity and mitigating the cold-start problem. Extensive experiments using benchmark datasets underscore its prowess compared to other leading collaborative filtering models, highlighting SAWE's potential to usher in transformative advancements in the field.

6.1.1 Preliminary

We outline some of the key topics for this subject across this section. Extended information is available at Section 2.1.4. The following are the definitions:

Bipartite Graph. A simple graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ is called bipartite if its vertex set $\mathcal{V}$ can be partitioned into two disjoint subsets of respective users and items, $\mathcal{U} \cup \mathcal{I}$, such that every edge $e \in \mathcal{E}$ has the form $e = (u, i)$ where $u \in \mathcal{U}$ and $i \in \mathcal{I}$. Accordingly, it can be defined as $\mathcal{B} = \{\mathcal{U}, \mathcal{I}, \mathcal{E}\}$.

Adjacency Matrix. The bipartite graph $\mathcal{B}$ has an adjacency matrix $\boldsymbol{A}$ that can be defined as:

$$\boldsymbol{A}_{N\times N} = \begin{bmatrix} 0 & \boldsymbol{R} \\ \boldsymbol{R}^\intercal & 0 \end{bmatrix}, \tag{6.1}$$

where $N \times N$ are dimensions for $\boldsymbol{A}$, $\boldsymbol{R}$ is an implicit feedback matrix with dimensions $|\mathcal{U}| \times |\mathcal{I}|$, $\boldsymbol{R}_{u,i} = 1$ if there is a relation between $u \in \mathcal{U}$ and $i \in \mathcal{I}$, otherwise $\boldsymbol{R}_{u,i} = 0$.

Laplacian Matrix. The Laplacian matrix is defined as $\boldsymbol{L} = \boldsymbol{I} - \boldsymbol{D}^{-1}\boldsymbol{A}$, where $\boldsymbol{I}$ is the identity matrix, and $\boldsymbol{D}^{-1}$ is the inverse of the diagonal degree matrix. The matrix $\boldsymbol{L}$ possesses a set of eigenvectors $\boldsymbol{U} = (\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_N)$, and a corresponding set of eigenvalues $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_N)$, which is a diagonal matrix composed of the eigenvalues.

Normalized Laplacian Matrix. the following equations represent the normalized Laplacian:

$$\boldsymbol{L}^{\eta} = \boldsymbol{U}\Lambda\boldsymbol{U}^{\boldsymbol{T}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{L}\mathbf{D}^{-\frac{1}{2}} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}} \tag{6.2}$$

Where $\mathbf{L}^{\eta}$ has a set of orthogonal eigenvectors $\mathbf{U} = (\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_N)$ and is accompanied by a set of eigenvalues $\widetilde{\mathbf{\Lambda}} = \text{diag}([\tilde{\lambda_1}, \tilde{\lambda_2}, \ldots, \tilde{\lambda}_N])$. Thus, it is often necessary to find its eigenvectors $\mathbf{U}$ and eigenvalues $\widetilde{\mathbf{\Lambda}}$. However, full computation can be resource-intensive, especially for large matrices. An efficient approach is to solve the following equation for a selected number of the smallest eigenvalues, denoted by Q, where Q is significantly less than N:

$$\mathbf{L}\mathbf{u}_i = \lambda_i\mathbf{u}_i \quad \text{for} \quad i = 1, \ldots, Q \tag{6.3}$$

In this equation, $\mathbf{u}_i$ represents the columns of $\mathbf{U}$, and λ_i are the smallest Q eigenvalues of the matrix $\mathbf{L}$. This selective computation is advantageous as it reduces the computational complexity and resources required. Various algorithms and techniques for efficiently finding the smallest eigenvalues and eigenvectors of a matrix are used, such as the Lanczos algorithm, Chebyshev polynomials, and Arnoldi iteration.

6.1.2 The problem and implications.

Given a user set $\mathcal{U}$ and an item set $\mathcal{I}$, for each user $u \in \mathcal{U}$ who interacted with at least an item $i \in \mathcal{I}$, we define set $\mathcal{I}_u^+ \subseteq \mathcal{I}$ as the set of interacted items while we define set $\mathcal{I}_u^- \subseteq \mathcal{I}$ as the set of remaining items. A user interaction with an item includes rating, clicking, or viewing an item. Implicit feed-back element $\boldsymbol{R}_{u,i} = 1$ if $i \in \mathcal{I}_u^+$. The objective of a CFRS model is to predict the value of $\hat{\boldsymbol{R}}_{u,j}$, where $j \in \mathcal{I}_u^-$, and recommend a ranked list of items from $\mathcal{I}_u^-$, associated with the most preferred values of $\hat{\boldsymbol{R}}_{u,j}$. CFRS models aim to learn $\hat{\boldsymbol{R}}_{u,j} = f(u, j|\Theta)$, where f is an interaction function to map u and j to a predicted preference score $\hat{\boldsymbol{R}}_{u,j}$. f learns the values of the model parameters set Θ through optimizing the selected objective function, Bayesian Personalized Ranking (BPR) in this research context. Therefore, we propose a novel Wavelet Collaborative Filtering model to address challenges three and four in Section 1.4, which are the cold start problem and the long-tail problem.

To address the challenges we've identified, we've developed a CFRS model that combines Graph Wavelet Transforms (GWT) and a Smoothing Layer. Our model successfully integrates Eigenvalue Filtering, Scale-Aware Graph Wavelets, and a Neural Multi-Layer Architecture into a Graph Convolutional Network (GCN). This integration improves how we learn from the spectral domain of user-item graphs. Here are the main contributions of our research:

- We've devised a novel technique named Scale-Aware Wave Graph Embedding (SAWE), which enhances our ability to interpret and learn from the spectral domains of bipartite graphs. SAWE represents a unique blend of spectral theory and deep learning principles, which we've ingeniously integrated within the context of CFRS.

- SAWE goes beyond conventional methods by employing a cluster-based approach to eigenvalues. This approach allows for nuanced and adaptive treatment of eigenvalues. The technique involves applying a custom smoothing function to each cluster of eigenvalues, with the process optimized based on the Signal-to-Noise Ratio (SNR). This introduces a unique blend of adaptive and cluster-specific smoothing.

- Another key contribution is the use of SNR for optimizing the smoothing factors, which facilitates a more informed assessment of the quality of filtered eigenvalues and enables better identification of important information within clusters. This results in superior eigenvalue filtering, capturing patterns at both macro and micro levels, a significant improvement over traditional uniform or fixed filtering methods.

- We've shown that SAWE performs exceptionally well in addressing major CFRS challenges, such as improving the diversity of recommendations and dealing with the cold-start problem. These results show the flexibility and effectiveness of our model in different recommendation scenarios.

- We've carried out extensive experiments to evaluate the performance of our model compared to other leading CFRS models. These experiments used several benchmark datasets that had been pre-processed. The detailed comparisons highlight the strengths of our proposed method and its potential to make a significant contribution to the field.

In summary, these contributions demonstrate the uniqueness and importance of our research and how it could influence future developments in the field of CFRS and spectral graph theory.

6.1.3 Graph Wavelet Transform (GWT)

GWT has a similar task to GFT as it transforms the graphs signal from the spatial domain into spectral domain. Similarly to GFT, bases are defined as $\boldsymbol{\Psi}_s = (\boldsymbol{\psi}_{s1}, \psi_{s2}, .., \psi_{sN})$. $\boldsymbol{\Psi}_s$ can be written as

$$\boldsymbol{\Psi}_s = \boldsymbol{U}\mathrm{G}_s\boldsymbol{U}^\intercal \tag{6.4}$$

where $\boldsymbol{U}$ is Laplacian eigenvectors while the matrix G_s =diag $(g(s\lambda_1), .., g(s\lambda_N))$ is used for scaling. In this research context, $g(s\lambda_n) = e^{-\lambda_n s}$. s is a given hyper-parameter used to control the search range for similar nodes [48, 155]. To illustrate, higher value of s facilitates capturing patterns of larger neighborhoods. Using graph wavelets as bases, GWT of a signal $\boldsymbol{x}$ on a graph is defined as $\hat{\boldsymbol{x}} = \boldsymbol{\Psi}_s^{-1}\boldsymbol{x}$ while the inverse of GWT is presented as $\boldsymbol{x} = \boldsymbol{\Psi}_s\hat{\boldsymbol{x}}$. Based on GFT , we write graph convolution as:

$$\begin{bmatrix}\boldsymbol{x}^u\\ \boldsymbol{x}^i\end{bmatrix} *_{\mathcal{G}} \boldsymbol{y} = \boldsymbol{\Psi}_s((\boldsymbol{\Psi}_s^{-1}\boldsymbol{y}) \odot (\boldsymbol{\Psi}_s^{-1}\begin{bmatrix}\boldsymbol{x}^u\\ \boldsymbol{x}^i\end{bmatrix})) = \boldsymbol{\Psi}_s(\hat{\boldsymbol{y}} \odot \begin{bmatrix}\hat{\boldsymbol{x}}^u\\ \hat{\boldsymbol{x}}^i\end{bmatrix}) \tag{6.5}$$

GWT improvements to graph convolution. GWT has the following advantages over GFT:

- Sparsity: table 2.2 shows that the GWT matrix $\boldsymbol{\Psi}_s$ is considerably sparse for all datasets when compared with GFT matrix $\boldsymbol{U}$, which makes most operations related to GWT more efficient computationally.

- Localized convolution: Consider a node as an origin point from which each wavelet $\boldsymbol{\Psi}_s$ disseminates akin to a signal [48, 155, 156]. Localization bolsters the efficacy of graph analysis in various tasks, such as semi-supervised learning on graphs, and augments the interpretability of graph analysis [48, 156].

- Enhanced scalability and adaptability: In earlier methodologies, the shortest path distance metrics were employed to tailor the quest for analogous nodes. In contrast, graph wavelets adopt a more adaptable strategy through the utilization of the hyper-parameter s [48, 155]. Notably, the value of s is autonomous of the graphs' statistical properties, eigenvectors, or eigenvalues. Concurrently, it exerts a uniform influence on all eigenvalues.

6.2 Method

The following subsections represents the research methodology:

6.2.1 SAWE Design Principles

The SAWE model has been designed to effectively tackle challenges commonly associated with spectral collaborative filtering through wavelet transformation, eigenvalues clustering, and eigenvalues filtering. These challenges broadly cover data sparsity, localized convolution, scalability, and adaptability. We have meticulously designed experiments to assess SAWE's performance in addressing these issues. The combination of SAWE's architecture, notably its use of convolution processes in a Graph CNN to establish the Graph Wavelet Neural Network (GWNN), is central to its efficiency. This section sheds light on SAWE's foundational design elements and the corroborating experiments.

Addressing Data Sparsity and Scalability

SAWE employs the Cluster-Based Eigenvalue Filtering approach to address issues of data sparsity and to ensure optimal scalability. The multi-layered structure of the GWNN ensures that it processes data seamlessly across layers, even in situations with limited data. Our testing approach involved utilizing datasets with varying levels of sparsity. Comparing SAWE's outcomes with other reference models on these datasets revealed its proficiency in handling sparse datasets and its ability to scale.

Dealing with Localized Convolution

For localized convolution, SAWE integrates Scale-Aware Graph Wavelets. This allows the GWNN to extract detailed insights from individual nodes across multiple scales. Our testing strategy involved grouping users based on their interaction intensity, which effectively varied the data's granularity. Observing SAWE's performance in such settings demonstrated its aptitude in managing data across varied scales within a graph.

Ensuring SAWE's Adaptability

A distinguishing feature of SAWE is its integration of wavelets within a Neural Multi-Layer Model. By emphasizing progressive learning through layers and incorporating the Leaky Rectified Linear Activation function, SAWE ensures adaptability. To validate this, we altered user-item interactions to emulate varied contexts and analyzed SAWE's behavior across different neural network layers. These evaluations were aimed at gauging SAWE's adaptability to new datasets and its efficiency across different model structures.

6.2.2 Cluster-Based Eigenvalue Filtering

Cluster-Based Eigenvalue Filtering is a technique designed to process a set of Laplacian eigenvalues, denoted as $\widetilde{\mathbf{\Lambda}} = \mathrm{diag}(\tilde{\lambda}_1, \ldots, \tilde{\lambda}_N)$. The aim is to perform selective smoothing on these eigenvalues based on their characteristics. The method involves several steps. First, it clusters the Laplacian eigenvalues in $\widetilde{\Lambda}$ to group similar ones together. After the eigenvalues are clustered, a cluster-sensitive smoothing function is applied to each group, with different levels of smoothing within each cluster. The level of smoothing is then optimized for each cluster based on the Signal-to-Noise Ratio (SNR). Finally, the technique outputs the filtered eigenvalues, represented as $\gamma_{11}, \gamma_{12}, \ldots, \gamma_{ZN}$, as well as their inverses, denoted as $\gamma_{11}^{-1}, \gamma_{12}^{-1}, \ldots, \gamma_{ZN}^{-1}$, where Z represents the respective cluster number.

Step 1: Clustering the Laplacian Eigenvalues

In this step, the Laplacian eigenvalues $\widetilde{\Lambda} = \mathrm{diag}(\tilde{\lambda}_1, \ldots, \tilde{\lambda}_N)$. are clustered into Z clusters using the KMeans algorithm. KMeans works by finding Z centroids in the eigenvalue space such that the sum of the Euclidean distances of each eigenvalue to its closest centroid is minimized [157]. The optimization problem can be formally defined as:

$$\min_{C_1,\ldots,C_Z} \sum_{i=1}^{Z} \sum_{\tilde{\lambda}_j \in C_i} \left\| \tilde{\lambda}_j - \mu_i \right\|^2, \tag{6.6}$$

where C_i represents the set of eigenvalues in the ith cluster, and μ_i is the centroid of the ith cluster.

The optimal number of clusters Z is determined using the silhouette score, which measures the cohesion and separation of the clusters [158]. The silhouette score $s(\tilde{\lambda}_i)$ for each eigenvalue is given by [158]:

$$s(\tilde{\lambda}_j) = \frac{b(\tilde{\lambda}_j) - a(\tilde{\lambda}_j)}{\max a(\tilde{\lambda}_j), b(\tilde{\lambda}_j)}, \tag{6.7}$$

where $a(\tilde{\lambda}_j)$ and $b(\tilde{\lambda}_j)$ are functions that compute the mean intra-cluster distance and the mean nearest-cluster distance, respectively. To illustrate, $a(\tilde{\lambda}_j)$ is the mean distance of $\tilde{\lambda}_j$ to all other points in the same cluster, and $b(\tilde{\lambda}_j)$ is the mean distance of $\tilde{\lambda}_j$ to all points in the nearest cluster. The number of clusters Z is selected such that the mean silhouette score across all eigenvalues is maximized.

Step 2: Applying Cluster-Sensitive Smoothing Function

The cluster-based approach allows us to apply a Cluster-Sensitive Smoothing Function, which generates weights for the Laplacian eigenvalues within each cluster. This function is defined as:

$$f(s_i, \tilde{\lambda}_{ij}) = e^{s_i \tilde{\lambda}_{ij}}, \tag{6.8}$$

where s_i is a smoothing factor and $\tilde{\lambda}_{ij}$ represents the jth Laplacian eigenvalue in the ith cluster. The exponential function acts as the transfer function, and s_i lies within an interval $[\underline{s}, \overline{s}]$, representing the lower and upper limits of the smoothing factor.

Using this function, the filtered and inverse-filtered eigenvalues for each cluster are computed as follows:

$$\gamma_{ij} = \tilde{\lambda}_{ij} \cdot f(s_i, \tilde{\lambda}_{ij}), \tag{6.9}$$

$$\gamma_{ij}^{-1} = \tilde{\lambda}_{ij} \cdot \frac{1}{f(s_i, \tilde{\lambda}_{ij})} \tag{6.10}$$

In this equation, the eigenvalues are scaled by the inverse of the smoothing function.

Step 3: Optimizing the Smoothing Factors Based on Signal-to-Noise Ratio (SNR)

During this step, we utilize the Signal-to-Noise Ratio (SNR) to evaluate the caliber of the eigenvalues following the application of filtering techniques [159]. SNR serves as a measurement for determining the strength of the desired signal compared to the interference caused by background noise. In this particular context, it gauges the average size of the eigenvalues in relation to how much they fluctuate within a cluster. Essentially, SNR is calculated by dividing the mean value of the eigenvalues by their average magnitude. (μ_{γ_i}) to their standard deviation (σ_{γ_i}) for a set of filtered eigenvalues γ_i within cluster i:

$$\text{SNR}(\gamma_i) = \frac{\mu_{\gamma_i}}{\sigma_{\gamma_i}}, \tag{6.11}$$

where

$$\mu_{\gamma_i} = \frac{1}{M} \sum_{j=1}^{M} \gamma_{ij}, \tag{6.12}$$

$$\sigma_{\gamma_i} = \sqrt{\frac{1}{M} \sum_{j=1}^{M} (\gamma_{ij} - \mu_{\gamma_i})^2}. \tag{6.13}$$

The goal is to optimize the smoothing factor s_i to maximize the SNR of the filtered eigenvalues. This is equivalent to minimizing the negative of the maximum SNR:

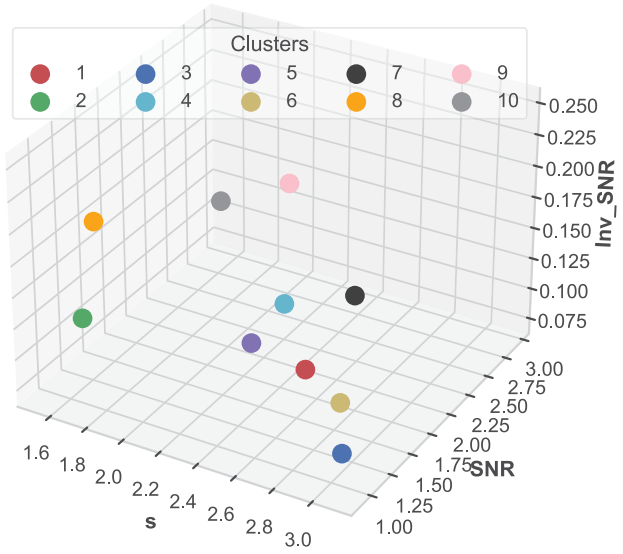
$$s_i^* = \arg \min_{s_i \in [\underline{s}, \overline{s}]} - \max(\text{SNR}(\gamma_{ij}), \text{SNR}(\gamma_{ij}^{-1})), \tag{6.14}$$

knowing that $\text{SNR}(\gamma_{ij}^{-1})$ is not the direct inverse of $\text{SNR}(\gamma_{ij})$. Through the Cluster-Based Eigenvalue Filtering process, the Laplacian eigenvalues are clustered, smoothed, and optimized in a way that is sensitive to their characteristics, enhancing their quality for further analysis or processing.

The 3D scatter plot (Figure 6.1(a)) illustrates the optimization process, highlighting the scale awareness of clusters through different "s" values. By assigning distinct smoothing factors to each cluster, the technique captures the varying characteristics and importance of clusters at different scales.

The Heatmap (Figure 6.1(b)) demonstrates how the method captures both larger-scale patterns and finer-scale details within the dataset. The colors of the filtered and inverse-filtered eigenvalues indicate the captured information, showcasing the effectiveness of the Cluster-Based Eigenvalue Filtering technique.

By combining these visualizations, a comprehensive understanding of the optimization process and the technique's ability to capture both Global and local patterns in the analyzed dataset can be obtained.



(a) Visualization of Cluster-Based Eigenvalue Filtering and Scale Awareness of Clusters in ML-1M Dataset

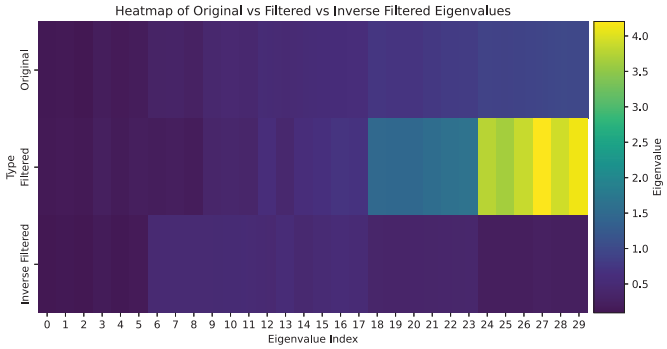


Figure 6.1: (b) Heatmap for 30 randomly generated eigenvalues.

6.2.3 Adaptive Spectral Graph Wavelets

Adaptive spectral graph wavelet basis, $\boldsymbol{\Psi}_\gamma$, is obtained using the Cluster-Sensitive Smoothing Function, defined as:

$$\boldsymbol{\Psi}_\gamma = \boldsymbol{\Phi}\boldsymbol{G}_\gamma\boldsymbol{\Phi}^\top, \quad \boldsymbol{\Psi}_\gamma^{-1} = \boldsymbol{\Phi}\boldsymbol{G}_\gamma^{-1}\boldsymbol{\Phi}^\top, \tag{6.15}$$

where $\boldsymbol{G}_\gamma$ is a diagonal matrix with filtered eigenvalues as its diagonal entries:

$$\boldsymbol{G}_\gamma = \text{diag}(\gamma_{11}, \gamma_{12}, \ldots, \gamma_{ZN}), \tag{6.16}$$

and $\boldsymbol{G}_\gamma^{-1}$ is the diagonal matrix of the inverse of these filtered eigenvalues:

$$\boldsymbol{G}_\gamma^{-1} = \text{diag}\left(\gamma_{11}^{-1}, \gamma_{12}^{-1}, \ldots, \gamma_{ZN}^{-1}\right). \tag{6.17}$$

6.2.4 Neural Multi-Layer Model

Replacing $\boldsymbol{\Psi}_s$ with $\boldsymbol{\Psi}_\gamma$ and $\boldsymbol{\Psi}_s^{-1}$ with $\boldsymbol{\Psi}_\gamma^{-1}$, Equation 6.5 is represented as:

$$\boldsymbol{\Psi}_\gamma((\boldsymbol{\Psi}_\gamma^{-1}\boldsymbol{y}) \odot (\boldsymbol{\Psi}_\gamma^{-1}\begin{bmatrix}\boldsymbol{x}^u\\ \boldsymbol{x}^i\end{bmatrix})) = \boldsymbol{\Psi}_\gamma(\hat{\boldsymbol{y}} \odot \begin{bmatrix}\hat{\boldsymbol{x}}^u\\ \hat{\boldsymbol{x}}^i\end{bmatrix}), \tag{6.18}$$

Simplifying the convolution operation further, we obtain:

$$\begin{bmatrix}\hat{\boldsymbol{x}}^u\\ \hat{\boldsymbol{x}}^i\end{bmatrix} = LReLU(\boldsymbol{\Psi}_\gamma\Lambda\boldsymbol{\Psi}_\gamma^{-1}\begin{bmatrix}\boldsymbol{x}^u\\ \boldsymbol{x}^i\end{bmatrix}\boldsymbol{P}), \tag{6.19}$$

where graph signals are $\boldsymbol{x}^u \in \mathbb{R}^{|\mathcal{U}|\times S}$ and $\boldsymbol{x}^i \in \mathbb{R}^{|\mathcal{I}|\times S}$ with S channels, and the learned signal is $\hat{\boldsymbol{x}}^u \in \mathbb{R}^{|\mathcal{U}|\times H}$ and $\hat{\boldsymbol{x}}^i \in \mathbb{R}^{|\mathcal{I}|\times H}$ with H filters. $\boldsymbol{P} \in \mathbb{R}^{S\times H}$ is the parameters matrix.

By incorporating the convolution operation into a Graph CNN, we establish a Graph Wavelet Neural Network (GWNN) with an l-layer structure:

$$\begin{bmatrix}\boldsymbol{x}_l^u\\ \boldsymbol{x}_l^i\end{bmatrix} = LReLU\left(\boldsymbol{\Psi}_\gamma\Lambda\boldsymbol{\Psi}_\gamma^{-1}\begin{bmatrix}\boldsymbol{x}_{l-1}^u\\ \boldsymbol{x}_{l-1}^i\end{bmatrix}\boldsymbol{P}_{l-1}\right), \quad l = 1..L \tag{6.20}$$

where $LReLU$ denotes Leaky Rectified Linear Activation (LReLU) function. The equation is performing a graph convolution followed by a Leaky ReLU activation, and then normalizing the batch with Batch Normalization. The input for the first layer, $\begin{bmatrix}\boldsymbol{x}_0^u\\ \boldsymbol{x}_0^i\end{bmatrix}$, and $\boldsymbol{P}_0$ are randomly initialized.

The outputs of each layer are concatenated into latent factors for users and items:

$$\begin{bmatrix}\boldsymbol{w}^u\\ \boldsymbol{w}^i\end{bmatrix} = \left[\begin{bmatrix}\boldsymbol{x}_0^u\\ \boldsymbol{x}_0^i\end{bmatrix} \mid \ldots \mid \begin{bmatrix}\boldsymbol{x}_L^u\\ \boldsymbol{x}_L^i\end{bmatrix}\right], \tag{6.21}$$

where $\begin{bmatrix}\boldsymbol{w}^u\\ \boldsymbol{w}^i\end{bmatrix}$ is the latent matrix. We also follow Dropout regularization, which is a technique to prevent overfitting in neural networks. During training, dropout randomly sets a fraction of input units to 0 at each update.

For the loss function, we employ Bayesian Personalized Ranking (BPR), which has shown better performance than alternative approaches such as Hinge loss:

$$\mathcal{L} = \min_{\boldsymbol{w}^u, \boldsymbol{w}^i} \sum_{(o,c,c') \in \mathrm{D}} - \log_e \sigma(\boldsymbol{w}_o^{u\top} \boldsymbol{w}_c^i - \boldsymbol{w}_o^{u\top} \boldsymbol{w}_{c'}^i) + \lambda(||\boldsymbol{w}^u||_2^2 + ||\boldsymbol{w}^i||_2^2), \tag{6.22}$$

where $\boldsymbol{w}_o^u$ represents the o-th column of $\boldsymbol{w}^u$, c and c' are the indices for viewed and non-viewed items by user o, respectively, and λ is a regularization hyper-parameter.

6.3 Experiments

In this section, we carry out experiments to evaluate the efficacy of the proposed SAWE schema against a group of baselines employing multiple benchmark datasets. As we present and dissect our experimental outcomes, our objective is to address the ensuing chapter queries:

- **CHQ1:** How does SAWE's performance measure up against the state-of-the-art models in the realm of recommendation systems?
- **CHQ2:** What is the impact of different user activity levels on the quality of the recommendations generated by SAWE?
- **CHQ3:** How effectively does SAWE tackle and alleviate the challenges associated with cold-start recommendations?
- **CHQ4:** Is SAWE model overfitted to specific hyper-parameter settings, such as the increasing number of epochs and the depth of the neural network?

The following represent details of experiments:

Details of Implementation. To keep the comparison fair, we stick to the same experiment setup as mentioned in [28, 66, 160, 161]. We measure the performance of SAWE alongside several CFRS methods, knowing that further information about the baselines are available at Section 2.1.13. Furthermore, information about metrics are available at Section 2.1.11. We used three datasets in experiments, which are explained in Section 2.1.12. The MovieLens-1M dataset includes only users and movies with a minimum of ten interactions to ensure robust data quality. Similarly, the Gowalla dataset is filtered to include only users and locations with at least fifteen interactions. The Amazon Musical Instruments and Amazon Arts, Crafts, and Sewing datasets also follow this pattern, retaining only users and products that have engaged in at least fifteen interactions.

We extensively pre-processed the benchmark datasets to ensure fair comparisons among the experimental baselines. Furthermore, we divide the datasets into 80% for training and 20% for testing. Following the advice in [48], we change the elements of the spectral graph wavelet matrix Ψ_γ and its inverse Ψ_γ^{-1} to 0 if they are tinier than 1e-7 to make the calculations faster. For deciding the number of times to run through the dataset, we use an early stopping approach [23].

6.3.1 Performance across Users with Different Interaction Levels

To tackle **CHQ2**, we employ the ACS and ML-1M datasets for illustration purposes by dividing the test sets into four groups, each containing users who have interacted with a certain number of items within a specific range (e.g., the first group includes users who have interacted with less than 25 items). As illustrated in Figure 6.2, the proposed model excels over the baselines in all user groups. Specifically, SAWE performs better than HMLET in all user groups on the ACS dataset, indicating that our method enhances recommendation performance for users who have interacted with a relatively small number of items. The improvements are also noticeable on the ML-1M dataset, as both HMLET and SimGCL baselines perform almost on par when users have

interacted with items between 50 and 100, whereas our proposed approach yields a noticeable improvement over the remaining baselines. A similar trend can be observed on the ACS dataset, indicating performance consistency across various datasets. This better performance is largely attributed to the fact that unlike the graph Fourier transform, graph wavelets are sparse, multiscale, and localized in the graph node domain. Table 1(in Appendix .1) shows the experiment's approach to handling the challenges described in Section 1.4.

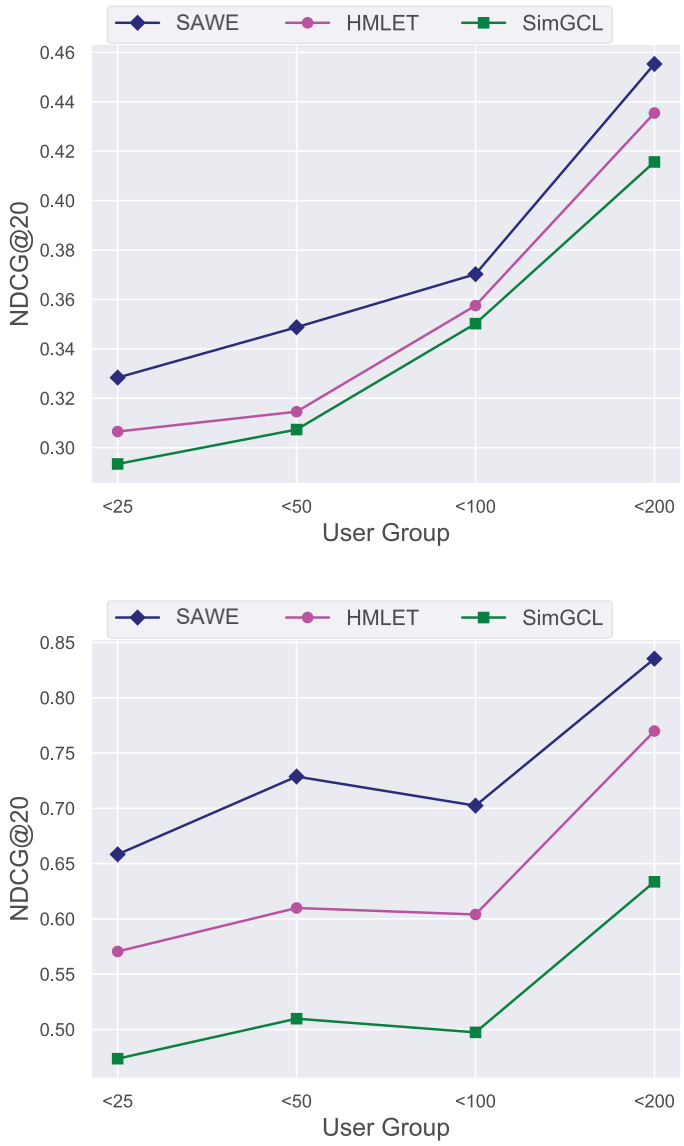


Figure 6.2: Performance comparison across users with varying interaction levels on the ML-1M (top) and ACS (bottom) datasets.

6.3.2 Results for Cold-Start Recommendations

CFRS is naturally affected by the problem of not having enough data, which often happens when users interact with or rate a small variety of items. This widespread issue is known as the cold-start problem, and it usually restricts the ability of a recommendation system to provide high-quality suggestions. To measure the effectiveness of the model we suggest in reducing the cold-start problem (**CHQ3**), we analyze the quality of recommendations made by SAWE for cold-start users compared to the current HMLET model. This analysis is done on the ACS dataset by creating different training sets with various levels of data scarcity. This goal is

met by changing the number of items that each user is connected to, within the set $3, 5, 7, 9, 12$. The results are shown in Table 6.1, indicating that both SAWE and HMLET are affected by the cold-start problem, resulting in a decline in performance as the number of cold-start users reduces. However, SAWE performs significantly better than HMLET in both Recall@20 and NDCG@20. On average, SAWE outperforms HMLET with relative improvements of 41.74% and 37.58% in Recall@20 and NDCG@20, respectively. Therefore, SAWE shows its ability to tackle the cold-start issue in CFRS, which can be credited not only to its adaptability but also to its skill in handling graph signals across different areas and scales. Table 1(in Appendix .1) shows the experiment's approach to handling the challenges described in Section 1.4.

Table 6.1: Performance Comparison in terms of Recall@20 and NDCG@20 in the sparse training sets by varying the number of items associated with each users in the ACS dataset. The average results are reported and bold numbers indicate better performance.

	# Items	3	5	7	9	12
Recall@20	HMLET	0.049	0.058	0.065	0.073	0.083
	SAWE	**0.076**	**0.083**	**0.092**	**0.099**	**0.109**
	Improvement	55.10%	43.10%	41.54%	35.62%	31.33%
NDCG@20	HMLET	0.058	0.064	0.071	0.078	0.089
	SAWE	**0.084**	**0.091**	**0.098**	**0.106**	**0.113**
	Improvement	44.83%	42.19%	38.03%	35.90%	26.97%

6.3.3 Ablation Studies

In this subsection we attempt to answer **CHQ1** and **CHQ4**. The depth of the network and the count of epochs are pivotal in determining the efficacy of the recommendation performance delivered by the proposed schema. We evaluate them in the next two sub-sections.

Performance comparison at different layers. The oscillation issue is a common challenge in practical settings, especially when dealing with deep networks. This issue manifests as unpredictable shifts in a model's performance when the network's layer count changes. To explore this phenomenon, we put SAWE to the test, evaluating its performance against a set of reputable benchmark methods including Graph Convolutional Matrix Completion (GC-MC) [162], NGCF [61], LightGCN [23], DGCF [67], and HMLET [18].

We refer to the experimental method previously described [67], conducting our tests on the ML-1M5 dataset. This customized version of the ML-1M dataset retains only vital ratings and omits users and items with less than ten interactions [67]. The refined dataset provides a more focused and reliable platform for training and testing the models. This helps limit noise and irrelevant data, offering a more accurate assessment of the models' performance across different network depths and their ability to handle the oscillation issue.

We also evaluated SAWE-S, a reduced version of SAWE. In this variant, the Cluster-Sensitive Smoothing Function is substituted with unfiltered eigenvalues.

The summarized results, as presented in Table 6.2, indicate that SAWE supersedes the other benchmark methods in most layers. Importantly, SAWE demonstrates consistent and impressive performance across all network depths, highlighting its ability in tackling the oscillation issue.

Obviously, the absence of adaptive learning features in SAWE-S resulted in diminished performance, even when compared to LightGCN. We will delve deeper into this topic in the subsequent subsection, further illustrating the importance of these features.

In contrast, models like GC-MC, NGCF, and LightGCN, exhibit a downward trend in performance as the network depth increases, which may suggest an overfitting tendency at higher depths. Meanwhile, models like DGCF, and HMLET show more stable performance, reflecting their unique ways of handling network depth

changes and the oscillation issue. Table 1(in Appendix .1) shows the experiment's approach to handling the challenges described in Section 1.4.

In summary, this experiment underscores the significance of model selection, data preprocessing, and network depth consideration in refining CFRSs' performance. However, given that the ML-1M5 dataset has undergone extensive preprocessing, care should be taken when generalizing these findings to more balanced, real-world datasets.

Table 6.2: Performance comparison of SAWE and baselines with increasing network depth on the ML-1M5 dataset in terms of NDCG@20. Boldface numbers indicate better performance.

	Layer 1	Layer 2	Layer 3	Layer 4
GC-MC	0.2574	0.2602	0.2328	0.2646
NGCF	0.2107	0.2111	0.1767	0.1759
LightGCN	0.2930	0.2844	0.2543	0.1968
DGCF	0.3037	0.3041	0.3027	0.3012
HMLET	0.3237	0.3291	0.3221	0.3189
SAWE-S	0.2749	0.2789	0.2752	0.2709
SAWE	**0.3329**	**0.3386**	**0.3307**	**0.3292**

Influence of number of Epochs. Figure 6.3 illustrates a performance comparison based on Recall@20 on the ML-1M5 dataset, by varying the number of epochs for SAWE, SAWE-S, and HMLET. It is clear from the Figure 6.3 that SAWE consistently outperforms HMLET across different epoch counts. Taken together, the results indicate that SAWE is proficient in learning more refined embeddings for users and items, exhibiting superior performance compared to HMLET. In contrast, SAWE-S demonstrates modest performance when compared to SAWE.

The limited performance of SAWE-S is largely due to the absence of the Cluster-Sensitive Smoothing Function. This lack of smoothing causes the model to excessively focus on the graph's local characteristics, leading to a risk of overfitting and limiting its ability to generalize. Furthermore, it might increase the complexity of the model and heighten the computational demands, due to an augmented emphasis on high frequency components. Essentially, this absence of smoothing undermines the model's ability to modulate the prominence of certain graph frequencies, thus complicating the training process. This particular issue has been addressed in the literature [48, 156, 163]. Table 1(in Appendix .1) shows the experiment's approach to handling the challenges described in Section 1.4.

6.4 Conclusions

In this research, we have introduced the SAWE approach, a novel method to process Laplacian eigenvalues in a graph signal processing context. This method represents a considerable advancement in the field of spectral-based collaborative filtering, first explored by SpectralCF in 2018.

The SAWE approach includes several innovative steps compared to previous research. It features a cluster-based approach to eigenvalues, introducing an adaptive and sophisticated smoothing process. The application of a cluster-sensitive smoothing function within each group, optimized based on the Signal-to-Noise Ratio (SNR), provides an adaptive and cluster-specific smoothing, previously not seen in the field.

The optimization of the smoothing factors based on the SNR is a particularly novel contribution, allowing for a better evaluation of the quality of the filtered eigenvalues and more informative cluster identification. This leads to enhanced eigenvalue filtering, capable of capturing both larger and finer-scale patterns in the

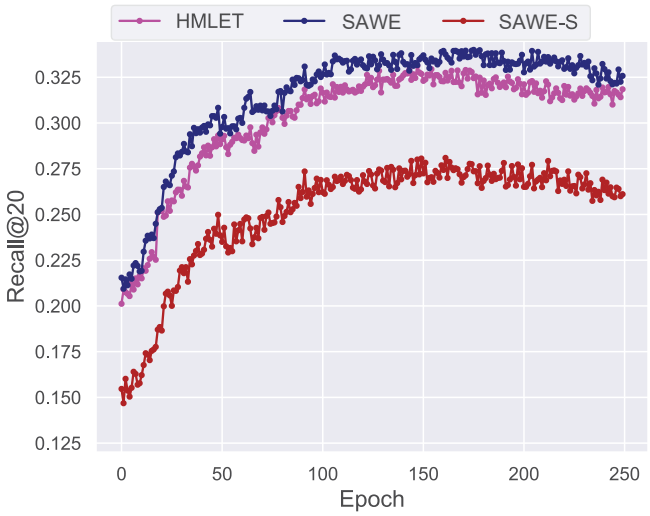


Figure 6.3: Performance comparison of SAWE, SAWE-S, and HMLET with increasing number of epochs on the ML-1M5 dataset.

dataset. This has led to a more comprehensive and adaptive filtering approach than the commonly used uniform or fixed filtering methods.

Additionally, we connected the filtered eigenvalues to the design of graph wavelet neural networks (GWNN), thus providing a novel neural multi-layer model that leverages the cluster-based filtering technique. This novel application of eigenvalue filtering in GWNN design distinguishes this work from previous research.

In summary, our SAWE approach brings notable advancements in the quality of filtered eigenvalues, and its potential applications, particularly in graph neural networks. This work offers a unique and efficient approach compared to traditional methods, with the potential to revolutionize the way spectral-based collaborative filtering is conducted in the future. Future work will be centered around furthering this method's scalability and adaptability to evolving user-item interaction patterns.

Chapter 7

Solution 4 - Adaptive Spectral Graph Wavelets for CFRSs

7.1 Introduction

This chapter is based on a published research paper [3] about a CFRS, called CAGLE, for **C**ollaborative Filtering with **A**daptive Spectral **G**raph Wave**LE**ts on graph-structured data. Table 1(in Appendix .1) shows the experiment's approach to handling the challenges described in Section 1.4.

CFRSs are instrumental in improving users' web browsing experience and retaining customers by providing tailored suggestions specific to the customer's needs and helping users find items of interest [164]. These systems provide valuable insight into consumers' purchasing behavior and browsing activity, and have proven to be effective not only at driving sales up in E-commerce, but also at increasing the number of people signing up for and using social media sites. When it comes to buying online, a convenient shopping experience that is tailored to consumers is probably the most important factor, with next day delivery is a standard now. Also, many consumers are turning to Internet shopping, particularly during pandemic lockdowns, opening up new opportunities for online retailers and other sellers of consumer goods.

Collaborative filtering is a widely used technique in RSs [165, 166], aiming at identifying user preferences using explicit feedback such as user ratings (e.g. 5-star or 10-point user rating systems) or implicit feedback such as user interactions with items (e.g. purchase, watch, listen, or click on an item). For the former, a vast majority of approaches rely on matrix factorization, which represents users and items in a low-dimensional latent space [167]. While explicit feedback includes explicit input by users regarding their interest in items, it is, however, not always available. By contrast, implicit feedback, in which user preferences are expressed via item interactions, is more common and practical, but poses challenges for CFRSs. In CFRS with implicit feedback, which is the focus of our work, the lack of interaction of a user with an item does not necessarily mean that an item is irrelevant for the user. A popular approach for collaborative filtering with implicit feedback is Bayesian personalized ranking (BPR) based matrix factorization [168]. BPR is a learning-to-rank method that produces a personalized ranking list of recommendations using a pairwise loss function, and assumes that observed user-item interactions should be ranked higher than the unobserved ones. While matrix factorization models are simple and effective [167, 167, 169], they are inherently linear and hence unable to capture the nonlinear structure of the implicit feedback data.

In recent years, a plethora of deep neural networks and graph-based recommendation models have been applied to the CFRS setting, showing improved recommendation performance [28, 61, 66, 170–175]. This boost in performance, compared to linear recommendation models, is attributed in large part to the fact that deep neural networks learn nonlinear features obtained from stacking up feature extractors, which are passed through non-linear activation functions, and hence capture more complex patterns. Also, the success of

deep neural networks in CFRSs and other downstream tasks has been greatly accelerated by using graphics processing units, which have become the platform of choice for training large, complex learning systems.

Previous work [28] on graph-based recommendation via graph convolutional networks has recently demonstrated that CFRS based on spectral graph theory is capable of discovering deep connections between users and items, and hence helps alleviate the cold-start problem. The general idea in spectral graph theory is that to any graph we may associate a corresponding matrix, which records information about its structure through the matrix spectrum. While the spectral collaborative filtering model provides good recommendation performance [28], its convolution operation is, however, defined in terms of the graph Fourier bases, which yield localization only in the spectral domain. In this chapter context, we introduce an adaptive graph wavelet basis that captures the graph's global structure and yields localization of graph signals in both spatial and spectral domains. The proposed CFRS approach leverages graph neural networks in the spectral graph theoretic setting, and employs a power transform to stabilize the variance of the graph frequencies, which are attenuated while learning the embeddings of users and items. The main contributions of this work can be summarized as follows:

- We introduce an adaptive convolutional filter using spectral graph wavelets and a power transform, yielding localization of graph signals in both spatial and spectral domains.
- We develop a deep recommendation model (CAGLE) for efficiently learning low-dimensional embeddings of users and items in a bipartite graph by leveraging the properties of spectral graph wavelets.
- We demonstrate through extensive experiments the superior performance of CAGLE over state-of-the-art baselines on several benchmark datasets.

The rest of this chapter is structured as follows. The related work section looks at previous research on the subject. The methodology section presents a deep recommendation model based on adaptive spectral graph wavelets and describes the main components of the proposed CFRS framework for implicit data. The experimental results section highlights the performance of the recommendation model on real-world benchmark datasets. Lastly, the conclusion section summarizes the findings and proposes directions for future research.

7.2 Related Work

The basic goal of CFRS is to generate personalized recommendations by leveraging historical data about interactions between users and items. To achieve this goal, a variety of CFRS approaches have been proposed for learning with implicit feedback.

Deep Neural Networks for Recommendation. Much of the recent work in CFRSs leverages deep learning [66, 170, 176, 177], which has shown remarkable capabilities in learning discriminative feature representations by extracting high-level features from data using multilayered neural networks. He *et al.* [66] introduce neural collaborative filtering (NCF), a deep neural network that integrates the linear matrix factorization with the non-linear multi-layer perceptron in an effort to learn the interaction function that maps model parameters to the predicted interaction score through nonlinear neural optimization in lieu of the inner product of latent factors. A major drawback of NCF is that the number of the model's parameters grows linearly with both the number of users and items. Liang *et al.* [170] extend variational autoencoders to collaborative filtering for implicit feedback using a generative model with a multinomial conditional likelihood function parameterized by a neural network. To learn the parameters of this generative model, the posterior distribution is approximated using variational inference. However, the choice of the prior may negatively impact the performance, especially when a lot of data is available for a user.

Graph Convolutional Networks for Recommendation. More recently, graph convolutional networks (GCNs) [148], which are an efficient variant of convolutional neural networks on graph-structured data, have proven to be useful in many graph analysis tasks, achieving state-of-the-art performance in various application domains, including recommendation [28, 61, 178]. Zheng *et al.* [28] propose spectral collaborative filtering (SpectralCF), a deep recommendation model based on spectral filtering of graph signals using the graph Fourier transform. SpectralCF formulates the relationships between users and items as a bipartite graph, and employs polynomial approximations of the spectral filters in order to help alleviate the learning complexity of the model [149]. Wang *et al.* [61] introduce neural graph collaborative filtering (NGCF), a GCN-based recommendation model that directly encodes the collaborative information of users by leveraging the high-order connectivity from user-item interactions via embedding propagation. A higher-order GCN-based approach for collaborative filtering is also presented in [178], where multiple stacked mixed-order GCN layers are used, followed by an average pooling layer for information fusion. He *et al.* [23] present a simplified NGCF model by leveraging the simple graph convolution [151], which successively removes the nonlinear activation functions and collapses the weight matrices between consecutive layers. In this simplified model, the low-dimensional embeddings of users and items are learned iteratively for a pre-defined number of power iteration steps, and then combined using a weighted average to obtain the final representations. Our proposed framework differs from existing collaborative filtering approaches in two main aspects. First, we design an adaptive filter by leveraging a power transform with the aim to stabilize the variance of graph frequencies in the spectral domain. Second, we develop a deep recommendation model that captures not only the graph's global structure, but also yields localization of graph signals in both spatial and spectral domains.

7.3 Method

In this section, we first introduce our notation and provide a brief background on spectral collaborative filtering, followed by the problem formulation of collaborative filtering with implicit feedback on bipartite graphs. A bipartite graph is a graph whose nodes can be partitioned into two disjoint sets such that there is no edge that connects nodes from the same set [179]. Then, we present the main building blocks of our proposed graph wavelet collaborative filtering framework, which is illustrated in Figure 7.1. Each layer of our model takes the initialized user/item embeddings as input and returns embeddings that are obtained by applying a layer-wise propagation rule. Then, the final user/items embeddings are acquired by concatenating the representations learned at different layers. Finally, we take the inner product to predict the preference of a user toward an item.

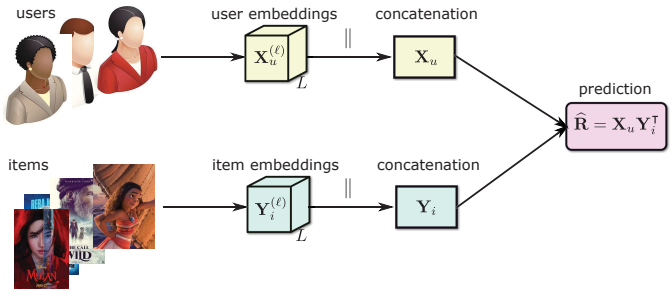


Figure 7.1: Schematic diagram of the proposed framework.

7.3.1 Preliminaries and Problem Statement

A bipartite user-item graph $\mathcal{G} = (\mathcal{U}, \mathcal{I}, \mathcal{E})$ consists of a node set $\mathcal{U} = \{u_1, \ldots, u_M\}$ of users, a node set $\mathcal{I} = \{i_1, \ldots, i_K\}$ of items, and an edge set $\mathcal{E}$. An edge $e = (u, i) \in \mathcal{E}$ indicates that item i is of interest to user u. The node sets $\mathcal{U}$ and $\mathcal{I}$ are disjoints, and their union $\mathcal{U} \cup \mathcal{I}$ consists of $N = M + K$ nodes. It is worth

mentioning that a bipartite graph is 2-colorable (i.e. each node can be assigned one of two colors) and has no odd length cycles (i.e. no cycles with odd number of edges). An illustrative example of a bipartite graph with 3 users and 4 items is depicted in Figure 7.2, which shows that item i_3 is more likely to be of interest to the cold-start (target) user u_1, as there exist two paths connecting them compared to a single path between u_1 and the other items.

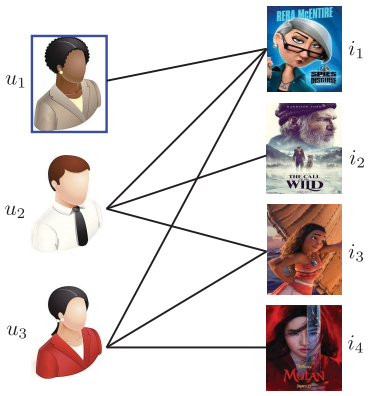


Figure 7.2: Illustration of a user-item bipartite graph.

Adjacency Matrix: The adjacency matrix of the bipartite user-item graph is defined as

$$\boldsymbol{A} = \begin{bmatrix} \mathbf{0} & \boldsymbol{R} \\ \boldsymbol{R}^\intercal & \mathbf{0} \end{bmatrix}, \tag{7.1}$$

where $\boldsymbol{R} = (\boldsymbol{R}_{ui})$ is an $M \times K$ implicit feedback matrix (also called biadjacency or user-item interaction matrix) representing the users' binary feedback (relevancy) for the items such that $\boldsymbol{R}_{ui} = 1$ if item i is an implicit interaction between user u and item i is observed; and 0 otherwise. Note that $\boldsymbol{R}_{ui} = 1$ does not necessarily mean that u likes i, just that the user may certainly have interest in the item. In a similar vein, $\boldsymbol{R}_{ui} = 0$ does not necessarily mean that u dislikes i, just that the user may be unaware of the item.

Normalized Laplacian Matrix: The normalized Laplacian matrix of the bipartite user-item graph is defined as

$$\boldsymbol{L} = \boldsymbol{I} - \boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}}, \tag{7.2}$$

where $\boldsymbol{D} = \mathrm{diag}(\boldsymbol{A}\mathbf{1})$ is the diagonal degree matrix, and $\mathbf{1}$ is an N-dimensional vector of all ones.

Graph Fourier Basis: Since the normalized Laplacian matrix is symmetric positive semi-definite [180], it admits an eigendecomposition given by $\boldsymbol{L} = \boldsymbol{\Phi}\boldsymbol{\Lambda}\boldsymbol{\Phi}^\intercal$, where $\boldsymbol{\Phi} = (\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_N)$ is an orthonormal matrix whose columns constitute an orthonormal basis of eigenvectors (called Graph Fourier basis) and $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_N)$ is a diagonal matrix comprised of the corresponding eigenvalues such that $0 = \lambda_1 \leq \cdots \leq \lambda_N = 2$. Note that $\lambda_N = 2$ only holds for bipartite graphs.

Graph Wavelet Basis: Spectral graph wavelets have shown to allow localization of graph signals in both spatial and spectral domains [48, 155, 181, 182]. Let $g_s(\lambda) = e^{-\lambda s}$ be the transfer function (also called frequency response) of the heat kernel with scaling parameter s. The spectral graph wavelet basis $\boldsymbol{\Psi}_s$ is defined as

$$\boldsymbol{\Psi}_s = \boldsymbol{\Phi}\boldsymbol{G}_s\boldsymbol{\Phi}^\intercal, \tag{7.3}$$

where $\boldsymbol{G}_s = g_s(\boldsymbol{\Lambda}) = \mathrm{diag}(g_s(\lambda_1), \ldots, g_s(\lambda_N))$ is a diagonal matrix. Note that $\boldsymbol{\Psi}_s$ is also referred to as the heat kernel matrix whose inverse $\boldsymbol{\Psi}_s^{-1}$ is obtained by simply replacing the scale parameter s with its negative value.

Spectral Collaborative Filtering: Given input user embedding matrix $\boldsymbol{X}_u^{(\ell)} \in \mathbb{R}^{M \times F_\ell}$ and input item embedding matrix $\boldsymbol{Y}_i^{(\ell)} \in \mathbb{R}^{K \times F_\ell}$ of the ℓ-th layer with F_ℓ feature maps, the output feature matrices $\boldsymbol{X}_u^{(\ell+1)}$ and $\boldsymbol{Y}_i^{(\ell+1)}$ of spectral collaborative filtering (SpectralCF) are obtained by applying the following layer-wise propagation rule [28]:

$$\begin{bmatrix} \boldsymbol{X}_u^{(\ell+1)} \\ \boldsymbol{Y}_i^{(\ell+1)} \end{bmatrix} = \sigma \left(\boldsymbol{\Phi} \widetilde{\boldsymbol{\Lambda}} \boldsymbol{\Phi}^\intercal \begin{bmatrix} \boldsymbol{X}_u^{(\ell)} \\ \boldsymbol{Y}_i^{(\ell)} \end{bmatrix} \boldsymbol{W}^{(\ell)} \right), \tag{7.4}$$

for $\ell = 0, \ldots, L-1$, where $\boldsymbol{W}^{(\ell)} \in \mathbb{R}^{F_\ell \times F_{\ell+1}}$ is a trainable weight matrix with $F_{\ell+1}$ feature maps, $\sigma(\cdot)$ is the point-wise sigmoid activation function, and $\widetilde{\boldsymbol{\Lambda}} = \boldsymbol{I} + \boldsymbol{\Lambda}$ is a diagonal matrix whose diagonal elements $\tilde{\lambda}_i$ are all positive.

Problem Statement. The objective of CFRS for implicit feedback is to leverage the user-item interaction matrix to estimate user-item preference scores for unobserved interactions. Denoting by $\mathcal{I}_u^+$ the set of items that user u has previously interacted with and $\mathcal{I}_u^- = \mathcal{I} \setminus \mathcal{I}_u^+$ its complement set, the goal of item recommendation with implicit feedback is to generate a top-K ranking of items from $\mathcal{I}_u^-$ that user u is most likely to prefer (e.g. purchase, watch, listen, or like). To address challenges (1, 3, 4, 7) in Section 1.4, we tailored the model evaluation to enhance performance for these specific challenges.

7.3.2 Proposed Method

Motivation. Spectral graph wavelet bases can yield localization of graph signals in both spatial and spectral domains whereas graph Fourier bases yield localization only in the spectral domain. In addition, graph wavelet bases are sparser than their graph Fourier counterparts, making them more computationally efficient. Unlike the graph Fourier basis, the graph wavelet basis is multiscale and hence can capture both local and global information from different graph neighborhoods by varying the value of the scaling parameter, which controls the amount of diffusion on the graph. For small values of the scaling parameter, the graph wavelet basis is determined by small (immediate) neighborhoods of a given node, reflecting local properties of the graph around that node. The larger the scaling parameter, the bigger is the support of the graph wavelet basis on the graph and hence the wavelet basis captures the graph's global structure.

Box-Cox Transformation. The spectrum of the normalized Laplacian matrix contains the structural information of the graph, as the multiplicity of the zero eigenvalue is exactly the number of connected components of a graph, while the second smallest eigenvalue is a global property that generally quantifies how well connected the graph is. In addition, the largest eigenvalue has an intuitive interpretation in the sense that how close is a graph is to being bipartite.

Each eigenpair $(\lambda_i, \boldsymbol{\varphi}_i)$ of the normalized Laplacian matrix can be interpreted in a similar fashion as the classical Fourier basis: the eigenvalues λ_i act as frequencies, while the eigenvectors $\boldsymbol{\varphi}_i$ play the role of the complex exponentials (sinusoidals). Since $\lambda_i = \boldsymbol{\varphi}_i^\intercal \boldsymbol{L} \boldsymbol{\varphi}_i$, we can interpret each eigenvalue as a measure of variation of its corresponding eigenvector. Eigenvectors associated with larger eigenvalues oscillate more rapidly, similar to the behaviors of classical Fourier basis functions in the Euclidean domain.

In order to stabilize the eigenvalues' variance, we can use a power transform [183], which essentially transforms non-normally distributed data to a set of data that has approximately normal distribution. More specifically, since the elements of $\widetilde{\boldsymbol{\Lambda}}$ are positive (i.e. $\tilde{\lambda}_i > 0$), we apply the Box-Cox transformation given by

$$\tilde{\lambda}_i^{(\kappa)} = \begin{cases} (\tilde{\lambda}_i^\kappa - 1)/\kappa & \text{if } \kappa \neq 0 \\ \log \tilde{\lambda}_i & \text{if } \kappa = 0 \end{cases} \tag{7.5}$$

where κ is the transformation parameter, which is estimated by maximizing the log-likelihood function using the boxcox function in SciPy, an open-source Python library for scientific and statistical computing. More

specifically, the SciPy boxcox function takes as input the elements of $\widetilde{\mathbf{\Lambda}}$ and returns both the Box-Cox transformed values and the optimal value for κ.

Adaptive Transfer Function. Denote by $\tilde{\mu}$ and $\tilde{\sigma}$ the sample mean and sample standard deviation of the Box-Cox transformed eigenvalues $\tilde{\lambda}_i^{(\kappa)}$. We define an adaptive transfer function of a low-pass filter with scaling parameter t as follows:

$$g_t(\tilde{\lambda}_i^{(\kappa)}) = \begin{cases} e^{-\tilde{\lambda}_i^{\kappa}(1+2t/c)} & \text{if } \tilde{\lambda}_i^{(\kappa)} < \tilde{\mu} \\ e^{-\tilde{\lambda}_i^{\kappa}(1+3t/c+\tilde{\sigma})} & \text{if } \tilde{\lambda}_i^{(\kappa)} \in (\tilde{\mu}, \tilde{\mu}+\tilde{\sigma}) \\ e^{-\tilde{\lambda}_i^{\kappa}(1+4t/c+\tilde{\sigma})} & \text{if } \tilde{\lambda}_i^{(\kappa)} \in (\tilde{\mu}+\tilde{\sigma}, \tilde{\mu}+2\tilde{\sigma}) \\ e^{-\tilde{\lambda}_i^{\kappa}(1+5t/c+\tilde{\sigma})} & \text{if } \tilde{\lambda}_i^{(\kappa)} > \tilde{\mu}+2\tilde{\sigma} \end{cases} \tag{7.6}$$

where $c = \sum_i \tilde{\lambda}_i^{(\kappa)}$ is the sum of Box-Cox transformed eigenvalues. This adaptive transfer function is illustrated in Figure 7.3.

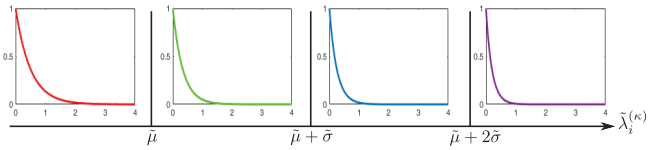


Figure 7.3: Adaptive transfer function plots for different scaling parameters from smaller to larger (left to right).

Adaptive graph wavelets. Using the adaptive transfer function, we define the adaptive spectral graph wavelet basis $\mathbf{\Psi}_t$ at scale t as follows:

$$\mathbf{\Psi}_t = \mathbf{\Phi} \boldsymbol{G}_t \mathbf{\Phi}^{\intercal}, \tag{7.7}$$

where $\boldsymbol{G}_t = g_t(\widetilde{\mathbf{\Lambda}}^{(\kappa)})$ and $\widetilde{\mathbf{\Lambda}}^{(\kappa)}$ is a diagonal matrix whose diagonal elements are the Box-Cox transformed eigenvalues. The proposed graph wavelet basis inherits many useful properties from the heat kernel, such as being multiscale and stable under perturbations of the graph. In addition, it is computationally efficient since we only need to compute the Q smallest eigenvalues and associated eigenvectors of the normalized Laplacian matrix due to the rapid decay of the transfer function, where $Q \ll N$. These eigenvalues/eigenvectors can be efficiently computed using the sparse eigensolver implemented in Python.

Learning Embeddings: Given input user embedding matrix $\boldsymbol{X}_u^{(\ell)} \in \mathbb{R}^{M \times F_\ell}$ and input item embedding matrix $\boldsymbol{Y}_i^{(\ell)} \in \mathbb{R}^{K \times F_\ell}$ of the ℓ-th layer with F_ℓ feature maps, the output feature matrices $\boldsymbol{X}_u^{(\ell+1)}$ and $\boldsymbol{Y}_i^{(\ell+1)}$ of CAGLE are obtained by applying the following layer-wise propagation rule:

$$\begin{bmatrix} \boldsymbol{X}_u^{(\ell+1)} \\ \boldsymbol{Y}_i^{(\ell+1)} \end{bmatrix} = \sigma\left(\mathbf{\Psi}_t \widetilde{\mathbf{\Lambda}} \boldsymbol{H}^{(\ell)} \mathbf{\Psi}_t^{-1} \begin{bmatrix} \boldsymbol{X}_u^{(\ell)} \\ \boldsymbol{Y}_i^{(\ell)} \end{bmatrix} \boldsymbol{W}^{(\ell)} \right), \tag{7.8}$$

where $\boldsymbol{W}^{(\ell)} \in \mathbb{R}^{F_\ell \times F_{\ell+1}}$ is a trainable weight matrix with $F_{\ell+1}$ feature maps, $\sigma(\cdot)$ is the point-wise sigmoid activation function, and $\boldsymbol{H}^{(\ell)}$ is a diagonal matrix given by

$$\boldsymbol{H}^{(\ell)} = \sigma\big(\boldsymbol{G}_t \odot \mathbf{\Theta}^{(\ell)}\big), \tag{7.9}$$

which controls how the graph frequencies (eigenvalues) are attenuated by learning a diagonal weight matrix $\mathbf{\Theta}^{(\ell)}$, and $\odot$ denotes the point-wise element (Hadamard) product.

The inputs of the first layer are side information matrices $\boldsymbol{X}_u^{(0)} \in \mathbb{R}^{M\times F}$ and $\boldsymbol{Y}_i^{(0)} \in \mathbb{R}^{K\times F}$, which are often randomly initialized with appropriate values generated from a zero mean Gaussian distribution in an effort to prevent the problem of exploding or vanishing gradients. For simplicity, we assume that the feature dimensions are equal for all layers, i.e. $F_\ell = P$ for all $\ell \geq 1$, with $P \ll \min(M, K)$.

Concatenation. We combine all layers by concatenating their outputs as follows:

$$\boldsymbol{X}_u = \Big\Vert_{\ell=0}^{L} \boldsymbol{X}_u^{(\ell)} \quad \text{and} \quad \boldsymbol{Y}_i = \Big\Vert_{\ell=0}^{L} \boldsymbol{Y}_i^{(\ell)} \tag{7.10}$$

resulting in a user embedding matrix $\boldsymbol{X}_u \in \mathbb{R}^{M\times(1+L)P}$ and an item embedding matrix $\boldsymbol{Y}_i \in \mathbb{R}^{K\times(1+L)P}$, where $\Vert$ denotes column-wise concatenation.

Model Prediction. The concatenated matrices $\boldsymbol{X}_u$ and $\boldsymbol{Y}_i$ of all layers' embeddings can be used as input for downstream tasks such as classification, clustering, and recommendation. Since the latter task is the focus of this chapter, we apply inner product to estimate the predicted items for a target user as follows:

$$\widehat{\boldsymbol{R}} = \boldsymbol{X}_u \boldsymbol{Y}_i^\intercal, \tag{7.11}$$

where $\widehat{\boldsymbol{R}} = (\widehat{\boldsymbol{R}}_{ui})$ is an $M \times K$ matrix of estimated user-item preference scores. Each entry $\widehat{\boldsymbol{R}}_{ui} = \boldsymbol{x}_u \boldsymbol{y}_i^\intercal$ is the rating score of item i being recommended to user u, where $\boldsymbol{x}_u$ and $\boldsymbol{y}_i$ denote the rows of $\boldsymbol{X}_u$ and $\boldsymbol{Y}_i$, respectively.

Loss function. The parameters of the proposed CAGLE model are learned by minimizing a regularized Bayesian personalized ranking (BPR) loss function [168] given by

$$\begin{aligned} \mathcal{L} = &- \sum_{(u,i,j)\in\mathcal{D}} \log \sigma(\widehat{\boldsymbol{R}}_{ui} - \widehat{\boldsymbol{R}}_{uj}) \\ &+ \frac{\eta}{2}(\|\boldsymbol{X}_u\|_F^2 + \|\boldsymbol{Y}_i\|_F^2), \end{aligned} \tag{7.12}$$

with respect to $\boldsymbol{W} = \{\{\boldsymbol{W}^{(\ell)}\}_{\ell=0}^{L-1}, \{\boldsymbol{\Theta}^{(\ell)}\}_{\ell=0}^{L-1}, \boldsymbol{X}_u^{(0)}, \boldsymbol{Y}_i^{(0)}\}$, a set of learnable parameters, over the pairwise training set

$$\mathcal{D} = \{(u, i, j) : (u, i) \in \mathcal{U} \times \mathcal{I}_u^+ \text{ and } (u, j) \in \mathcal{U} \times \mathcal{I}_u^-\},$$

where $\sigma(\cdot)$ is the sigmoid function, and η is a tuning parameter that controls the strength of the regularization term, which is added to impose smoothness constraint on the solutions and also to prevent over-fitting.

A training triple $(u, i, j) \in \mathcal{D}$, which consists of one user and two items, indicates that user u is assumed to prefer item i over item j. In addition, minimizing $\mathcal{L}$ is tantamount to determining whether the observed pair (u, i) should have a higher user-item preference score than the unobserved pair (u, j). Hence, the aim is to maximize the margin between the observed score $\widehat{\boldsymbol{R}}_{ui}$ and the unobserved score $\widehat{\boldsymbol{R}}_{uj}$, as shown in Figure 7.4. The loss function can be rewritten as

$$\begin{aligned} \mathcal{L} = &- \sum_{(u,i,j)\in\mathcal{D}} \log \sigma(\boldsymbol{x}_u \boldsymbol{y}_i^\intercal - \boldsymbol{x}_u \boldsymbol{y}_j^\intercal) \\ &+ \frac{\eta}{2}\left(\sum_{u=1}^{M} \|\boldsymbol{x}_u\|_2^2 + \sum_{i=1}^{K} \|\boldsymbol{y}_i\|_2^2\right), \end{aligned} \tag{7.13}$$

Using the derivative property $\sigma'(\boldsymbol{z}) = \sigma(\boldsymbol{z})(1 - \sigma(\boldsymbol{z}))$ of the sigmoid function $\sigma(\cdot)$, the partial derivatives of

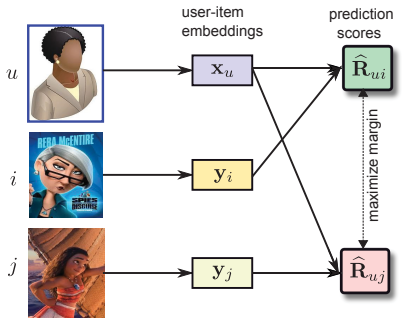


Figure 7.4: Illustration of training triple (one user and two items) in BPR.

$\mathcal{L}$ with respect to $\boldsymbol{x}_u$ and $\boldsymbol{y}_i$ are then given by

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{x}_u} = -(1 - \sigma(\boldsymbol{x}_u(\boldsymbol{y}_i - \boldsymbol{y}_j)^\intercal))(\boldsymbol{y}_i - \boldsymbol{y}_j) + \eta\, \boldsymbol{x}_u \tag{7.14}$$

and

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{y}_i} = -(1 - \sigma(\boldsymbol{x}_u(\boldsymbol{y}_i - \boldsymbol{y}_j)^\intercal))\boldsymbol{x}_u + \eta\, \boldsymbol{y}_i \tag{7.15}$$

Once the learning of the model's parameters is done, we can then compute a personalized ranked list for a user u based on the values of the estimated scores $\widehat{\boldsymbol{R}}_{ui}$ over all items.

7.4 Experiments

In this section, we conduct experiments to assess the performance of the proposed CAGLE framework in comparison with competing baseline models on several benchmark datasets. While presenting and analyzing our experimental results, we aim to answer the following chapter questions:

- **CHQ1:** How does CAGLE perform in comparison with state-of-the-art recommendation models?
- **CHQ2:** How does recommendation performance vary across users with different levels of activity?
- **CHQ3:** How does CAGLE alleviate the cold-start recommendation problem?
- **CHQ4:** Is CAGLE model overfitted to specific hyper-parameter settings, such as the number of epochs and the top k items?

The following represent details of the experiments:

Implementation Details. For fair comparison, we implement the proposed model and baseline methods using NeuRec[1], an open-source Python library for neural recommender models, and we follow similar experimental setup to the experiments in [28,66,160,161]. We measure the performance of CAGLE alongside several CFRS methods, knowing that further information about the baselines are available at Section 2.1.13. Furthermore, information about metrics are available at Section 2.1.11. Further information about the Datasets is available at Section 2.1.12. We utilized the Gowalla dataset [61], Amazon Musical Instruments dataset(MI) [154], and Amazon Arts, ML-1M dataset, Crafts and Sewing dataset(ACS) [154]. Moreover, we excluded the information about users and items that have less than twenty interactions for the Gowalla dataset. Furthermore, we excluded information with fewer than 10 interactions for the Amazon Musical Instruments dataset, and with fewer than twenty interactions for the Amazon Arts, Crafts, and Sewing dataset

[1]https://github.com/wubinzzu/NeuRec

We split the datasets into 80% training and 20% testing. As suggested in [48], we set the elements of the spectral graph wavelet matrix $\mathbf{\Psi}_t$ and its inverse $\mathbf{\Psi}_t^{-1}$ to 0 if they are smaller than 1e-7 in an effort to improve computational efficiency. For both CAGLE and SpectralCF, we set the embedding dimension to 64. We use the Glorot initialization for all models. We set the depth of the CAGLE network to three layers using the Adam optimizer with a batch size of 1024. For the number of epochs, we use an early stopping strategy [23]. The learning rate and the scale t of the graph wavelet basis are chosen via grid search with cross-validation over the sets $\{0, 0.001, 0.01, \ldots, 0.1\}$ and $\{0.1, 0.2, 0.4, \ldots, 2\}$, respectively.

7.4.1 Recommendation Performance

To answer **CHQ1**, we use figures 7.5-to-7.8 to show the performance comparison of CAGLE and baselines on all datasets in terms of Recall@k and NDCG@k by varying the value of k for top-k recommendation. As can be seen, the proposed approach yields the best recommendation performance in comparison with all baseline methods. In particular, we can observe that CAGLE performs better than DGCF and SimGCL on the large MI dataset. Moreover, CAGLE outperforms SpectralCF by a significant margin on the large MI dataset. This better performance of the proposed model is largely attributed to the fact that spectral graph wavelets yield localization of graph signals in both spatial and spectral domains, resulting in improved recommendation quality.

Table 1(in Appendix .1) shows the experiment's approach to handling the challenges described in Section 1.4.

7.4.2 Performance across Users with Different Interaction Levels

To answer **CHQ2**, we consider the ACS and ML-1M datasets for the sake of illustration by partitioning the test sets into four groups, each of which contains users who have interacted with a certain number of items in a specific range (e.g. the first group includes users who have interacted with less than 25 items). As shown in Figure 7.9, the proposed model outperforms the baselines on all user groups. In particular, CAGLE outperforms DGCF and SimGCL on all user groups in the ML-1M dataset, indicating that our approach improves recommendation performance for users who have interacted with a relatively small number of items. The improvements are particularly noticeable on the ML-1M dataset, as SimGCL baseline performs generally better than the other baselines when users have interacted with less than 25 items, whereas our proposed approach yields a relative improvement of approximately 7% over these competing baselines, which is a considerable improvement within these experimental settings. A similar trend can be observed on the ACS dataset, indicating performance consistency across various datasets. This better performance is largely attributed to the fact that unlike the graph Fourier transform, graph wavelets are sparse, multiscale, and localized in the graph node domain. Table 1(in Appendix .1) shows the experiment's approach to handling the challenges described in Section 1.4.

7.4.3 Cold-Start Recommendation Results

CFRS inherently suffers from the data sparsity problem, which often arises when users rate or interact with a small number of items. This common issue is known as the cold-start problem, which tends to hinder the ability of a CFRS to provide good quality recommendations. To assess the performance of our proposed model in mitigating the cold-start problem (**CHQ3**), we evaluate the quality of recommendations made by CAGLE for cold-start users in comparison with the recent Deoscillated Adaptive Graph Collaborative Filtering (DGCF) model [67] on the ACS dataset by creating different training sets with several degrees of sparsity. This is accomplished by varying the number of items, associated with each user, in the set $\{3, 5, 7, 9, 12\}$. The results are reported in Table 7.1, which shows that CAGLE, CAGLE-Fixed, and DGCF suffer from the cold-start problem, resulting in degraded performance as the number of cold-start users decreases. As expected, CAGLE-Fixed poor performance is apparent as it lacks flexibility and adaptive pattern learning mechanisms. Meanwhile, CAGLE significantly outperforms DGCF in terms of both Recall@20 and NDCG@20. On

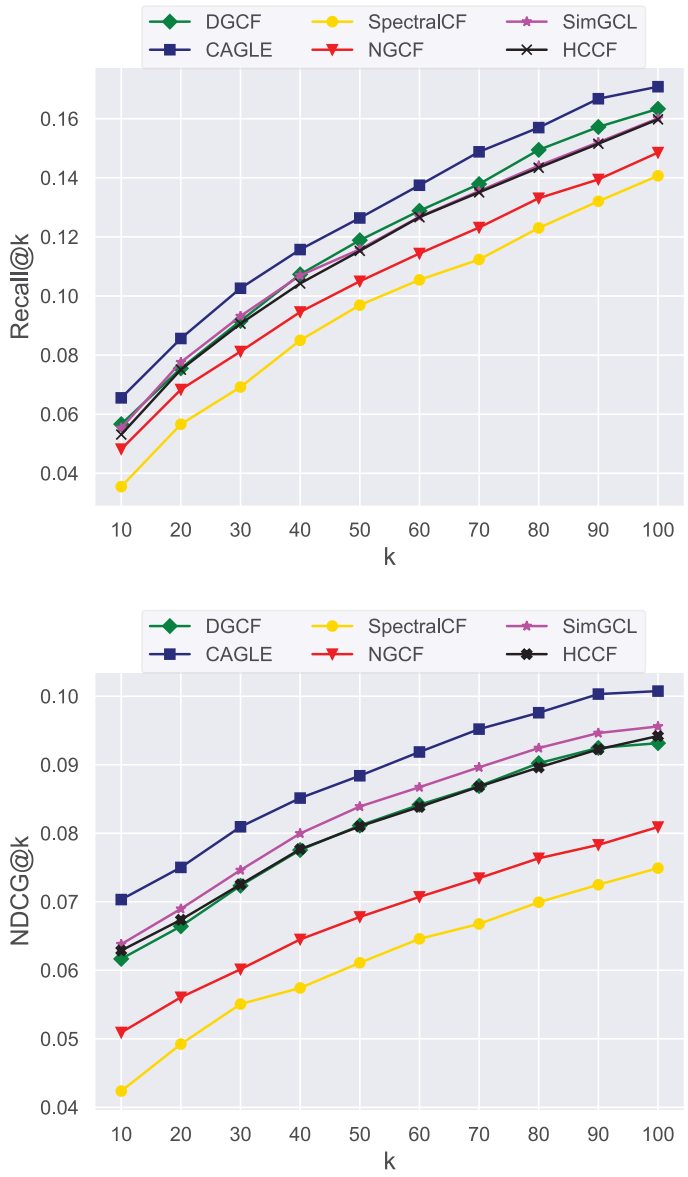


Figure 7.5: Performance comparison of CAGLE and baselines on the ACS dataset in terms of Recall@k and NDCG@k.

average, CAGLE outperforms DGCF by relative improvements of 43.56% and 47.63% in Recall@20 and NDCG@20, respectively. Hence, CAGLE is capable of helping alleviate the cold-start problem in CFRS, thanks in large part not only to its adaptive behavior, but also to its ability to localize graph signals in both spatial and spectral domains. Table 1(in Appendix .1) shows the experiment's approach to handling the challenges described in Section 1.4.

7.4.4 Ablation Studies

The network depth and number of epochs play an important role in the recommendation performance of the proposed framework.

Mitigating the Oscillation Problem. In practice, the oscillation problem often occurs when the network depth is relatively small [67], and refers to the trend of model performance as we vary the number of layers. To answer **CHQ4**, we compare the performance of CAGLE against several strong baseline methods,

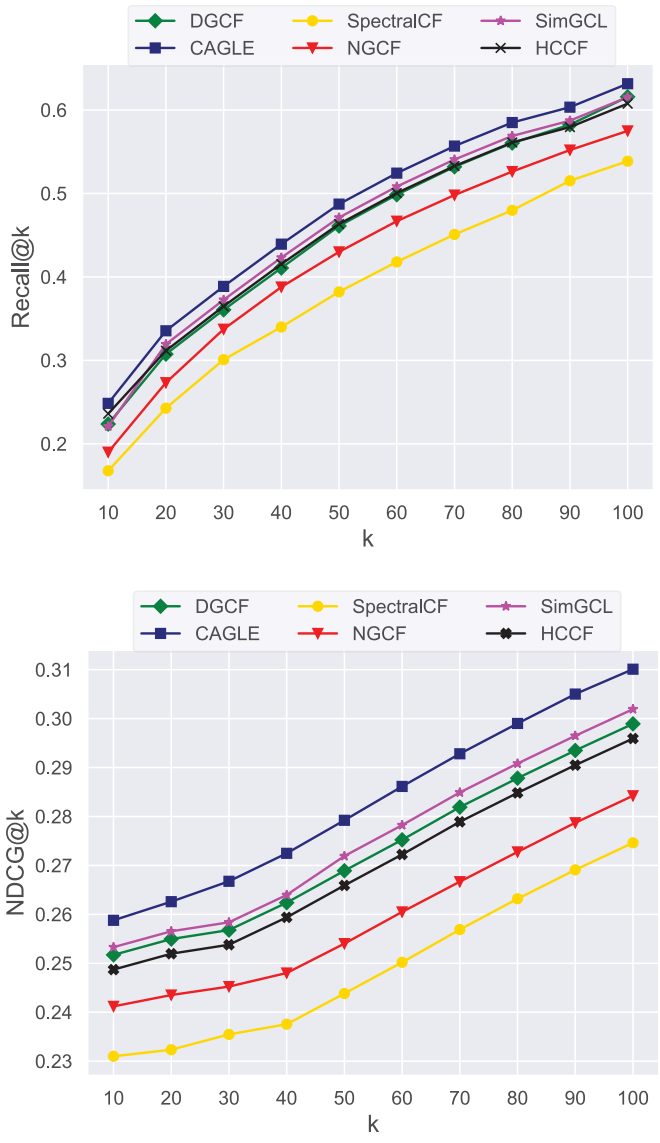


Figure 7.6: Performance comparison of CAGLE and baselines on the ML-1M dataset in terms of Recall@k and NDCG@k.

Table 7.1: Performance Comparison in terms of Recall@20 and NDCG@20 in the sparse training sets by varying the number of items associated with each users in the ACS dataset. The average results are reported and bold numbers indicate better performance.

	# Items	3	5	7	9	12
Recall@20	DGCF	0.017	0.021	0.024	0.026	0.029
	Fixed-CAGLE	0.009	0.011	0.014	0.019	0.023
	CAGLE	**0.025**	**0.029**	**0.034**	**0.038**	**0.042**
	Improvement	47.06%	38.09%	41.67%	46.15%	44.83%
NDCG@20	DGCF	0.015	0.019	0.022	0.024	0.026
	Fixed-CAGLE	0.010	0.013	0.017	0.020	0.022
	CAGLE	**0.023**	**0.028**	**0.032**	**0.035**	**0.038**
	Improvement	53.33%	47.37%	45.45%	45.83%	46.15%

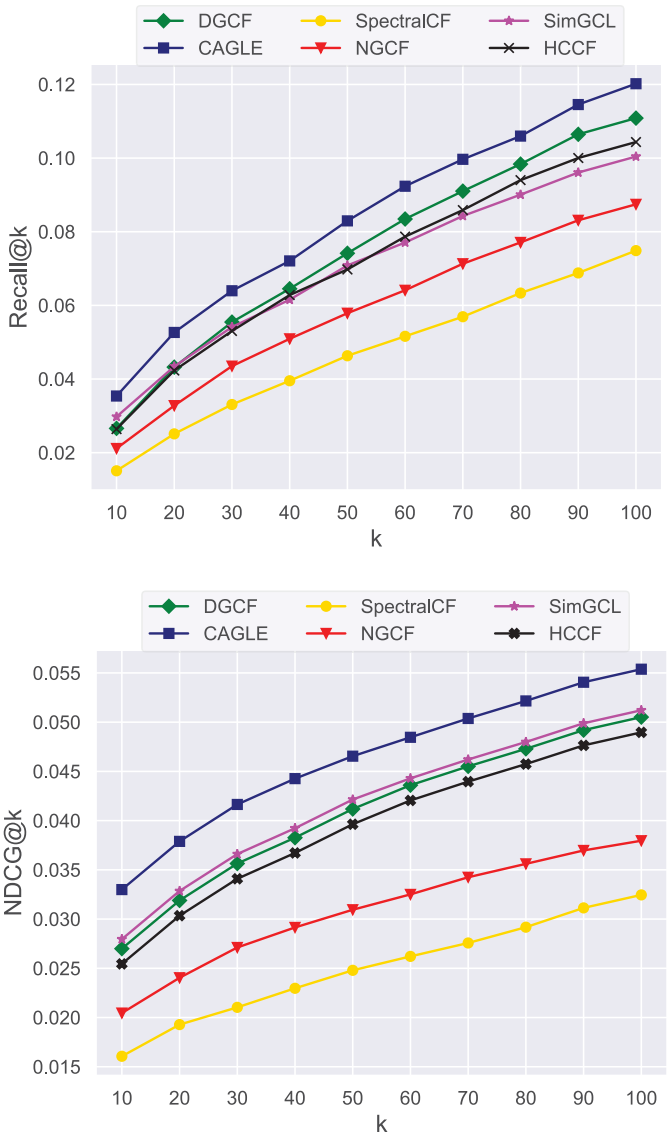


Figure 7.7: Performance comparison of CAGLE and baselines on the MI dataset in terms of Recall@k and NDCG@k.

including Graph Convolutional Matrix Completion (GC-MC) [162], HCCF [64], NGCF [61], LightGCN [23], DGCF [67]. Furthermore, we have customized a version of CAGLE (Fixed-CAGLE) that doesn't include the adaptive mechanisms of CAGLE, which are Adaptive Transfer Function and Adaptive graph wavelets. We follow the same experimental setup as [67] by conducting experiments on ML-1M5, a filtered version of ML-1M dataset, which is obtained by retaining only ratings equal to five [67]. The results are summarized in Table 7.2, which shows that CAGLE achieves better performance than all baselines across almost all layers, expect against DGCF with four layers. Fixed-CAGLE shows how modest is CAGLE performance without the adaptive mechanisms. Compared to the other baselines, CAGLE shows good performance on all first three network depths, indicating its ability to mitigate the oscillation problem. Table 1(in Appendix .1) shows the experiment's approach to handling the challenges described in Section 1.4.

Effect of Number of Epochs. Figure 7.10 shows the performance comparison of CAGLE, DGCF, and Fixed-CAGLE in terms of Recall@20 on the ML-1M5 dataset by varying the number of epochs. Notice that CAGLE outperforms DGCF when the number of epochs exceeds 30 while CAGLE outperforms Fixed-CAGLE on all

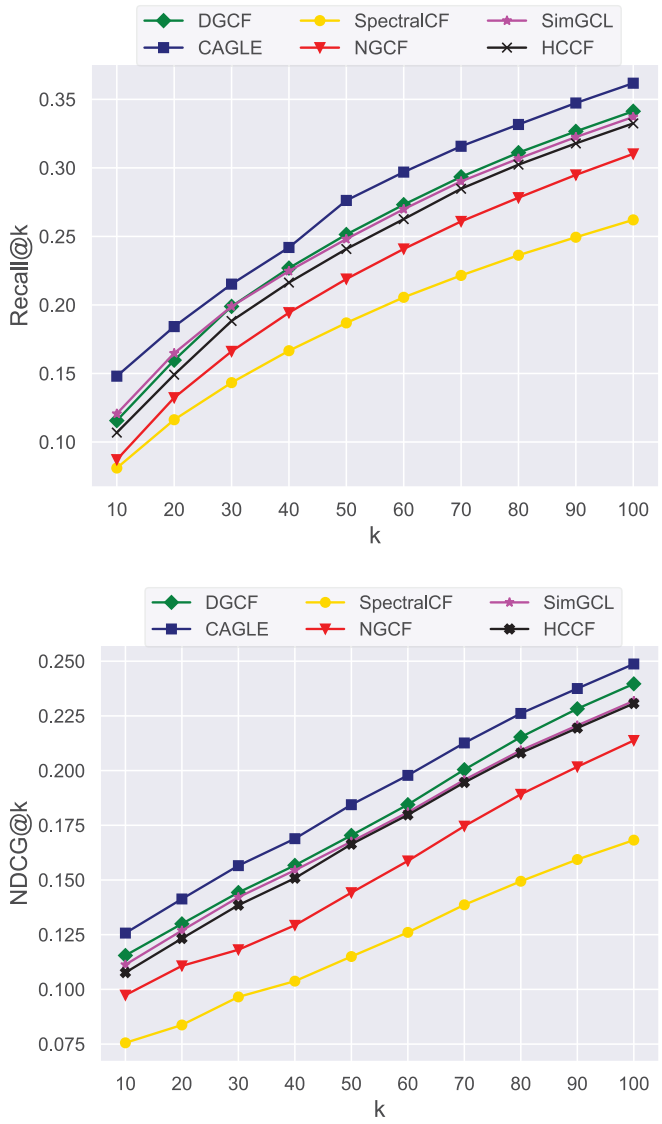


Figure 7.8: Performance comparison of CAGLE and baselines on the Gowalla dataset in terms of Recall@k and NDCG@k.

Table 7.2: Performance comparison of CAGLE and baselines with increasing network depth on the ML-1M5 dataset in terms of NDCG@20. Boldface numbers indicate better performance.

	Layer 1	Layer 2	Layer 3	Layer 4
GC-MC	0.2574	0.2602	0.2328	0.2646
NGCF	0.2107	0.2111	0.1767	0.1759
LightGCN	0.2930	0.2844	0.2543	0.1968
HCCF	0.2948	0.2971	0.2939	0.2895
DGCF	0.3037	0.3041	0.3027	0.3012
Fixed-CAGLE	0.2694	0.2741	0.2737	0.2692
CAGLE	**0.3086**	**0.3115**	**0.3182**	0.3012

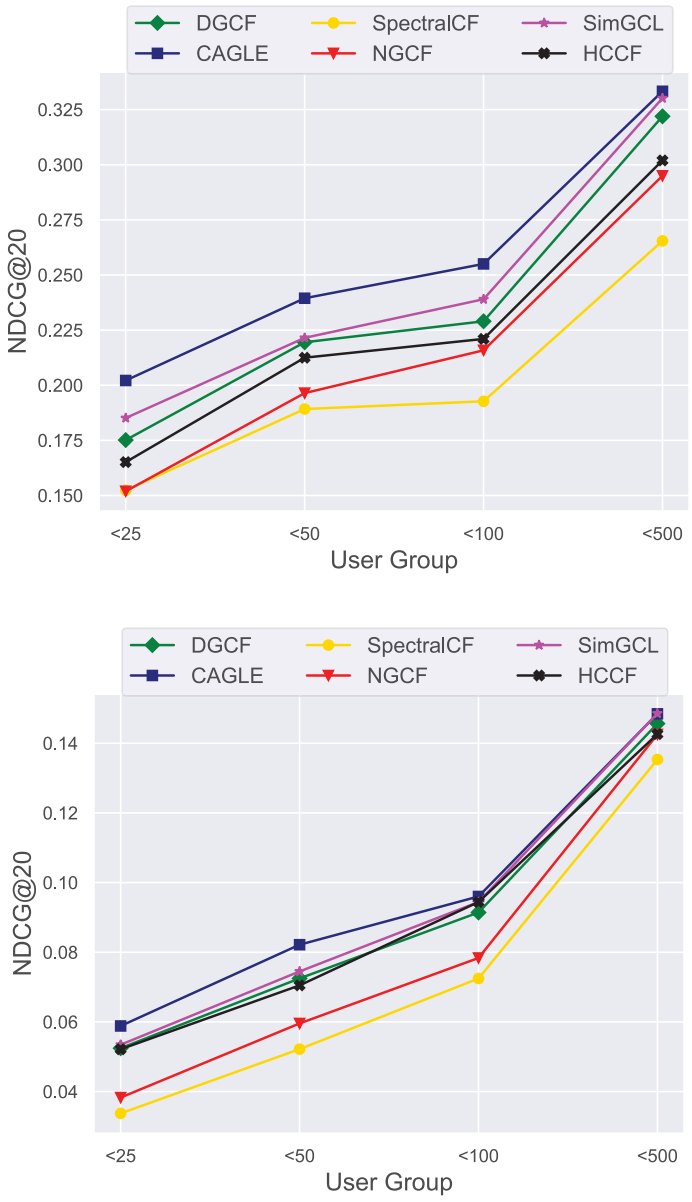


Figure 7.9: Performance comparison across users with varying interaction levels on the ML-1M (top) and ACS (bottom) datasets.

epochs. Overall, the proposed model yields better performance than DGCF and Fixed-CAGLE, indicating that CAGLE is able to learn more discriminative embeddings of users and items. Table 1(in Appendix .1) shows the experiment's approach to handling the challenges described in Section 1.4.

7.4.5 Statistical Significance Analysis

We perform statistical significance tests to compare CAGLE, DGCF, and SimGCL by employing multiple pairwise comparison analysis using Tukey's test. This test is employed to compare all possible pairs of means. The results of these comparisons are depicted in Figure 7.11, where 95% confidence interval plots are represented by horizontal lines. The red line signifies the comparison interval for the mean of CAGLE. This interval does not overlap with the comparison intervals for the means of DGCF and SimGCL, which are highlighted in grey and blue colors, respectively. As indicated by the horizontal dotted lines, DGCF and SimGCL have overlapping intervals, suggesting that there is no significant difference between their group

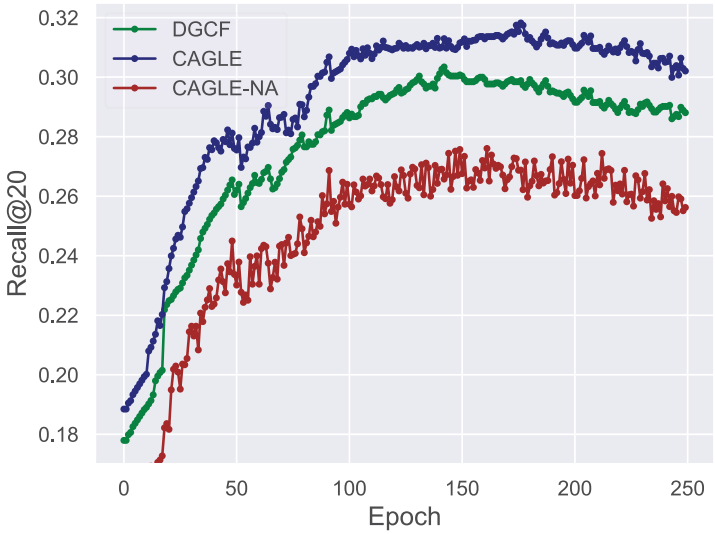


Figure 7.10: Performance comparison of CAGLE, Fixed-CAGLE, and DGCF with increasing number of epochs on the ML-1M5 dataset.

means. Conversely, CAGLE's interval does not overlap with those of SimGCL and DGCF, which implies a significant difference between the group means of CAGLE and the other two methods.

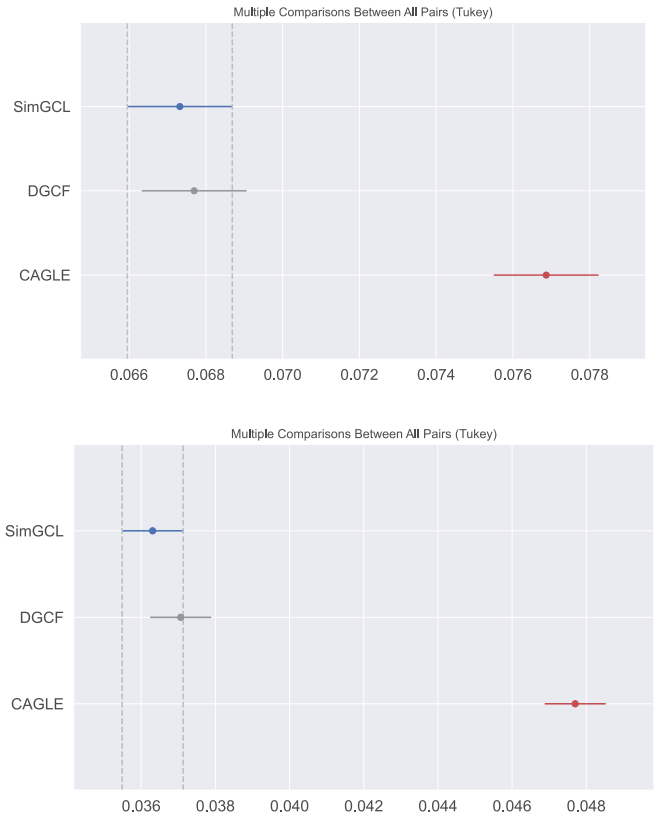


Figure 7.11: Pairwise multiple comparison between CAGLE, DGCF and SimGCL in terms of Recall@20 using Tukey's test on the ACS dataset (top) and MI dataset (bottom).

7.5 Conclusions

In this Chapter, we proposed a novel framework, named CAGLE, for CFRS with wavelets on graph-structured data. Our approach efficiently learns low-dimensional representations of users and items in the context of spectral graph theory. We first designed an adaptive graph convolutional filter using spectral graph wavelets. Then, we introduced a deep recommendation model, which yields localization of graph signals in both spatial and spectral domains. The proposed convolutional filter is multiscale and hence can capture both local and global information from different graph neighborhoods. Our experimental results on several real-world datasets demonstrated the effectiveness of the proposed model, showing performance gains over various baseline methods. For future work, we plan to incorporate high-order neighborhood information into the proposed framework.

Chapter 8

Conclusions and Future Works

This chapter presents the main contributions and findings of the research in developing improved CFRSs. The study addresses the importance of different quality attributes and their application in traditional engineering methods. Evaluation was performed following the ISO/IEC 25010:2011 standards. Conducted within the E-commerce domain, this research provides a basis for future studies and practical system innovations. These innovations were validated through empirical studies. The chapter concludes by discussing the scope and limitations of the research as well as suggesting directions for future work.

8.1 Main Contributions to Knowledge

This dissertation advances progress in CFRSs by integrating engineering methods with detailed evaluation standards. It addresses important gaps in improvement techniques and evaluation approaches, contributing significantly to academic research and practical applications in the fields of CFRSs and software engineering.

8.1.1 Critical Examination of Quality Attributes in CFRSs

The multidimensional essence of quality in CFRSs research mirrors a nuanced landscape of advancements and challenges. It is accuracy, diversity, privacy, and user satisfaction that play a central role in the qualities that generally add to CFRSs in terms of their general effectiveness and user-perceived nature.

Contribution The thesis critically looks at how, within the existing research, the said quality attributes were addressed and, therefore, gaps identified to propose methodologies for a more comprehensive enhancement of CFRSs' quality.

Impact This contribution opens new frontiers for academic research on holistic quality improvements, challenging industry practitioners to consider broad quality metrics in system development and enhancement.

8.1.2 Innovative Application of Traditional Engineering Methodologies

The very peculiar challenges of CFRSs—scalability, precision, and adaptability—call for new development and enhancement approaches. Classic engineering methodologies like System Block Diagrams, Functional Analysis, and TRIZ have structured strategies for problem-solving and have been utilized to a lesser extent in this context.

Contribution Adapting these methodologies to the CFRS domain provides a new framework and guideline for improvement in the design and functionality of these systems, addressing some of the key problems in system development.

Impact By setting a new paradigm for both academic researchers and industry practitioners, it paves the way for systematic improvements and innovative solutions in the field of recommendation systems.

8.1.3 Comprehensive Evaluation Through ISO/IEC 25010:2011 Standards

The adoption of standards of evaluation for the CFRS leads to broader and more comprehensive views on the quality of such systems, according to ISO/IEC 25010:2011. This approach is therefore made applicable from the traditional measures of accuracy to have more general quality characteristics such as usability, reliability, security, and performance efficiency.

Contribution By using the standards as a basis to have a multi-dimensional evaluation framework, this work calls for a holistic evaluation in enhancing the rigor and breadth in the evaluation of CFRSs.

Impact Academically, the model establishes the benchmark for quality assessment in CFRSs, while in industry, it provides a solid quality assurance framework and elevates user trust and satisfaction.

8.1.4 Empirical Validation in E-commerce Sector

The dynamic E-commerce sector provides a rich field to empirically test the practical applicability of the proposed framework. Empirical testing in this context is very important to verify and validate the real effectiveness of the developed methodologies and solutions.

Contribution This research now moves to empirical validation within the E-commerce domain to prove the strength of the framework in elevating the performance of the CFRS as well as improving the user engagement.

Impact These results provide concrete evidence of the effectiveness of the framework, informing further academic research and giving E-commerce platforms proven strategies to improve their CFRSs.

8.1.5 Foundation for Future Research and System Innovation

This would lay the basic foundation for upcoming research and innovation, without which the CFRSs would lack continuous improvement. The comprehensive framework that will be developed in this thesis not only addresses current challenges but sets up future research directions to further improve this area.

Contribution The thesis provides a solid ground for the development of further research and development in the field of CFRSs by offering a structured and validated approach to system enhancement and evaluation.

Impact It creates a cycle of academic research and industrial innovation by the adoption of advanced methodologies to stay ahead in a changing technological landscape.

8.1.6 Development and Validation of Practical Solutions

The development of practical solutions for addressing key challenges in CFRSs reflects the applied research approach of the thesis. Such solutions will further ensure increased accuracy of recommendations, improved scalability of the system, and, in the long run, maximum user satisfaction leading to business benefits.

Contribution The development and empirical validation of practical solutions are a milestone in the sense that new strategies for facing CFRSs' challenges are introduced, implemented, and validated with publication in reputable journals.

Impact Academically, this will contribute to the dialogue with implementable research results through peer-reviewed and published papers. With regard to the industry, it will offer approaches toward improved system personalization and effectiveness. In a nutshell, these contributions collectively make the literature and industrial practices of the field of CFRSs much enriched and lay the ground for future innovation and improvement of personalized recommendation technologies.

8.2 Scope and Limitations

Identifying the limitations of this study helps place its findings into perspective and opens avenues for further research. The scope and limitations of the work are outlined below to emphasize potential areas of

improvement and application:

1. **Focus on CFRS in E-commerce:** The study primarily focuses on CFRSs in the E-commerce sector, meaning its findings cannot be directly applied to other types of recommendation systems. Future work could aim to generalize these methods to other recommendation systems' types.

2. **Application Domain:** Since user preferences and behaviors differ across domains, the results of this study may not fully apply to other areas. While it is based on E-commerce, broader exploration of domain-specific adaptations could help expand its applicability.

3. **Application Level:** Although the experiments were designed to replicate real-world E-commerce scenarios, the solutions and framework have not been tested in real industry environments. Such testing could provide more practical validation and insights for refinement.

4. **Complexity in Integration:** Integrating engineering methodologies with CFRSs can be challenging, requiring expertise to adapt these methodologies. Efforts to reduce complexity could make the integration process more straightforward and enhance the approach's accessibility.

5. **Data Availability:** The methods heavily depend on the availability of detailed and relevant data. In cases where data is highly sparse or unavailable, the validity and generalization of findings could be limited. Developing methods to handle limited data scenarios could address this issue.

6. **Scope of Evaluation Standards:** While ISO/IEC 25010:2011 provides a broad evaluation framework, some quality aspects may have varying impacts on user satisfaction and system reliability. Including more specific and user-centered evaluation metrics could enhance the results.

These limitations offer insights into where improvements can be made and where the methods can be extended for greater relevance and practical application.

8.3 Future Works

CFRSs are still an ongoing development process, and improvement of the systems follow the same pattern. To continue making advancements, future work will extend the development of the current work in the following ways.

8.3.1 Expansion to Other Recommendation Types

Research so far has been leaning on CFRS; however, other recommendation types, such as content-based, and hybrid models, will provide a more comprehensive approach to recommendation systems. Future work must explore these methodologies to serve the diverse needs and preferences of users and thereby enhance the personalization and accuracy of recommendations.

8.3.2 Development of an Integrated Library of Tools

Besides, for the support of the frameworks developed in this thesis, there is a dire need for the availability of an integrated library of tools to be implemented and tested. This library will consist of software modules used in design, analysis, and testing of systems, hence providing a full toolbox for researchers or practitioners willing to embark on the task of developing or enhancing the existing CFRSs.

8.3.3 Inclusion of Other Quality Evaluation Frameworks

The provided standards by ISO/IEC 25010:2011 provide a good foundation to evaluate the quality of the systems; more quality evaluation frameworks can be included, which might give a diverse perspective on the performance of the systems. Future research should consider other well-documented standards and frameworks to provide a comprehensive analysis that covers all the aspects of quality in the systems.

8.3.4 Empirical Studies Across Varied Sectors

The application of the developed methodologies should, therefore, be empirically tested across many other sectors not limited to the E-commerce sector, to generalize the findings and verify their applicability across a varied domain. This would require conducting further studies in the multimedia, social networking, and education domain; the application of a CFRS could alter user experience and satisfactorily very highly.

8.3.5 Scalability and Adaptability

It is a well-agreed-upon theme for future work that systems should be enhanced in terms of scalability and adaptability. This is in view of the realization that the amount of data to be handled by systems will continue to grow, and thus, systems must be dynamic in meeting the user preference. This encompasses the development of new algorithms and architectures that will effectively process large datasets and dynamically adapt to user behavior and user preference modifications in real time.

8.3.6 Ethical Considerations and User Privacy

Since these recommendation systems are being integrated into more aspects of daily interaction, critical considerations in future research are the ethical side and user privacy to ensure that guidelines and mechanisms are proposed for the assurance of operations of the recommendation systems in a clear and user-privacy-preserving way.

8.3.7 Collaborative International Research Initiatives

There should be collaborative research efforts between the various international institutions to enhance a common global understanding and growth of a recommendation system. A partnership for this purpose, therefore, will entail sharing of insights, methodologies, and data and span different cultural and operational environments, and this invariably leads to enriched research outcomes having a global applicability.

These future works aim to not only advance the technical aspects of CFRSs but also address the broader implications of their deployment, ensuring that they serve the needs of users ethically and effectively while fostering global research collaboration.

References

[1] S. Abinaya and R. Ramya, "Enhancement in context-aware recommender systems–a systematic review," in *2024 Second International Conference on Emerging Trends in Information Technology and Engineering (ICETITE)*. IEEE, 2024, pp. 1–13.

[2] O. Alshareet and A. Awasthi, "Enhancing e-commerce recommendations with a novel scale-aware spectral graph wavelets framework," *International Journal of Data Science and Analytics*, pp. 1–14, 2023.

[3] O. Alshareet and A. Ben Hamza, "Adaptive spectral graph wavelets for collaborative filtering," *Pattern Analysis and Applications*, vol. 27, no. 1, p. 10, 2024.

[4] Y. Deldjoo, D. Jannach, A. Bellogin, A. Difonzo, and D. Zanzonelli, "Fairness in recommender systems: research landscape and future directions," *User Modeling and User-Adapted Interaction*, vol. 34, no. 1, pp. 59–108, 2024.

[5] V. L. Chetana and H. Seetha, "Handling massive sparse data in recommendation systems," *Journal of Information & Knowledge Management*, vol. 23, no. 03, p. 2450021, 2024.

[6] M. Ferrari Dacrema, P. Cremonesi, and D. Jannach, "Are we really making much progress? a worrying analysis of recent neural recommendation approaches," in *Proceedings of the 13th ACM conference on recommender systems*, 2019, pp. 101–109.

[7] M. Ferrari Dacrema, S. Boglio, P. Cremonesi, and D. Jannach, "A troubling analysis of reproducibility and progress in recommender systems research," *ACM Transactions on Information Systems (TOIS)*, vol. 39, no. 2, pp. 1–49, 2021.

[8] J. Lu, D. Wu, M. Mao, W. Wang, and G. Zhang, "Recommender system application developments: a survey," *Decision support systems*, vol. 74, pp. 12–32, 2015.

[9] L. Hardesty, "The history of amazon's recommendation algorithm," *Amazon Science*, vol. 22, 2019.

[10] K. Xu, H. Zhou, H. Zheng, M. Zhu, and Q. Xin, "Intelligent classification and personalized recommendation of e-commerce products based on machine learning," *arXiv preprint arXiv:2403.19345*, 2024.

[11] J. McAuley, C. Targett, Q. Shi, and A. Van Den Hengel, "Image-based recommendations on styles and substitutes," in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2015, pp. 43–52.

[12] C. C. Aggarwal *et al.*, *Recommender systems*. Springer, 2016, vol. 1.

[13] D. Tarakanova, "E-commerce entrepreneurship: a practical study of online shop development," 2023.

[14] J. Leskovec, A. Rajaraman, and J. D. Ullman, *Mining of massive data sets*. Cambridge university press, 2020.

[15] A. Ismail, "International marketing strategies in the celebrity cosmetics industry: a dual case study: Huda beauty and fenty beauty," 2018.

[16] P. Soto-Acosta, "Covid-19 pandemic: Shifting digital transformation to a high-speed gear," *Information Systems Management*, vol. 37, no. 4, pp. 260–266, 2020.

[17] G. Adomavicius and A. Tuzhilin, "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*, vol. 17, no. 6, pp. 734–749, 2005.

[18] T. Kong, T. Kim, J. Jeon, J. Choi, Y.-C. Lee, N. Park, and S.-W. Kim, "Linear, or non-linear, that is the question!" in *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, 2022, pp. 517–525.

[19] W. Wang, Y. Xu, F. Feng, X. Lin, X. He, and T.-S. Chua, "Diffusion recommender model," in *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2023, pp. 832–841.

[20] Z. Lin, C. Tian, Y. Hou, and W. X. Zhao, "Improving graph collaborative filtering with neighborhood-enriched contrastive learning," in *Proceedings of the ACM Web Conference 2022*, 2022, pp. 2320–2329.

[21] J. Yu, X. Xia, T. Chen, L. Cui, N. Q. V. Hung, and H. Yin, "Xsimgcl: Towards extremely simple graph contrastive learning for recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 2, pp. 913–926, 2023.

[22] C. Chen, M. Zhang, Y. Zhang, Y. Liu, and S. Ma, "Efficient neural matrix factorization without sampling for recommendation," *ACM Transactions on Information Systems (TOIS)*, vol. 38, no. 2, pp. 1–28, 2020.

[23] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "LightGCN: Simplifying and powering graph convolution network for recommendationg," in *Proc. International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020, pp. 639–648.

[24] I. Shenbin, A. Alekseev, E. Tutubalina, V. Malykh, and S. I. Nikolenko, "Recvae: A new variational autoencoder for top-n recommendations with implicit feedback," in *Proceedings of the 13th international conference on web search and data mining*, 2020, pp. 528–536.

[25] J. Wu, X. Wang, F. Feng, X. He, L. Chen, J. Lian, and X. Xie, "Self-supervised graph learning for recommendation," in *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, 2021, pp. 726–735.

[26] C. Liu, C. Yu, N. Gui, Z. Yu, and S. Deng, "Simgcl: graph contrastive learning by finding homophily in heterophily," *Knowledge and Information Systems*, vol. 66, no. 3, pp. 2089–2114, 2024.

[27] K. Mao, J. Zhu, J. Wang, Q. Dai, Z. Dong, X. Xiao, and X. He, "Simplex: A simple and strong baseline for collaborative filtering," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 1243–1252.

[28] L. Zheng, C.-T. Lu, F. Jiang, J. Zhang, and P. S. Yu, "Spectral collaborative filtering," in *Proceedings of the 12th ACM conference on recommender systems*, 2018, pp. 311–319.

[29] J. Lin, "The neural hype and comparisons against weak baselines," in *ACM SIGIR Forum*, vol. 52, no. 2. ACM New York, NY, USA, 2019, pp. 40–51.

[30] M. Ludewig and D. Jannach, "Evaluation of session-based recommendation algorithms," *User Modeling and User-Adapted Interaction*, vol. 28, pp. 331–390, 2018.

[31] Z. Chen, J. Piao, X. Lan, H. Cao, C. Gao, Z. Lu, and Y. Li, "Practitioners versus users: A value-sensitive evaluation of current industrial recommender system design," *Proceedings of the ACM on Human-Computer Interaction*, vol. 6, no. CSCW2, pp. 1–32, 2022.

[32] Z. Liu, Y. Li, N. Chen, Q. Wang, B. Hooi, and B. He, "A survey of imbalanced learning on graphs: Problems, techniques, and future directions," *arXiv preprint arXiv:2308.13821*, 2023.

[33] J. Bobadilla, J. Dueñas-Lerín, F. Ortega, and A. Gutierrez, "Comprehensive evaluation of matrix factorization models for collaborative filtering recommender systems," *arXiv preprint arXiv:2410.17644*, 2024.

[34] Z. Chen, W. Gan, J. Wu, K. Hu, and H. Lin, "Data scarcity in recommendation systems: A survey," *ACM Transactions on Recommender Systems*, 2024.

[35] "ISO/IEC 25010:2011 - Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models," https://www.iso.org/standard/35733.html, International Organization for Standardization, accessed on: 2024-02-01.

[36] A. Y. Doroshenko and I. Z. Achour, "Application of neuro evolution tools in automation of technical control systems," *Problems in Programming*, no. 1, pp. 16–25, 2021.

[37] W. C. Benton, "Machine learning systems and intelligent applications," *IEEE Software*, vol. 37, no. 4, pp. 43–49, 2020.

[38] K. R. Raghavendran and A. Elragal, "Low-code machine learning platforms: A fastlane to digitalization," in *Informatics*, vol. 10, no. 2. MDPI, 2023, p. 50.

[39] T. Zhang, *Mathematical analysis of machine learning algorithms*. Cambridge University Press, 2023.

[40] K. Zahoor, N. Z. Bawany, and T. Qamar, "Evaluating text classification with explainable artificial intelligence," *International Journal of Artificial Intelligence*, vol. 13, no. 1, pp. 278–286, 2024. [Online]. Available: https://ijai.iaescore.com/index.php/IJAI/article/download/22766/13828

[41] Z. Yang, "How to apply machine learning algorithms in artificial intelligence," *Advanced Educational Technology Research*, vol. 6, no. 1, pp. 401–410, 2023.

[42] B. Zlotin and A. Zusman, "Triz and practical artificial intelligence," *ITC 2023*, p. 16, 2023.

[43] B. Zlotin, A. Zusman, and M. Hohnjec, "Triz-based software as a front-end to ai chatbots," *ITC 2023*, p. 30, 2023.

[44] F. Ricci, L. Rokach, and B. Shapira, "Recommender systems: Techniques, applications, and challenges," *Recommender systems handbook*, pp. 1–35, 2021.

[45] S. Wu, F. Sun, W. Zhang, X. Xie, and B. Cui, "Graph neural networks in recommender systems: a survey," *ACM Computing Surveys*, vol. 55, no. 5, pp. 1–37, 2022.

[46] S. S. Khanal, P. Prasad, A. Alsadoon, and A. Maag, "A systematic review: machine learning based recommendation systems for e-learning," *Education and Information Technologies*, vol. 25, no. 4, pp. 2635–2664, 2020.

[47] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE signal processing magazine*, vol. 30, no. 3, pp. 83–98, 2013.

[48] B. Xu, H. Shen, Q. Cao, Y. Qiu, and X. Cheng, "Graph wavelet neural network," *arXiv preprint arXiv:1904.07785*, 2019.

[49] D. Kim and A. Oh, "How to find your friendly neighborhood: Graph attention design with self-supervision," *arXiv preprint arXiv:2204.04879*, 2022.

[50] S. N. Gondaliya and K. Amin, "Solutions to the cold start problem in recommender system: A survey," *Journal of Software Engineering Tools & Technology Trends*, vol. 5, no. 3, pp. 14–21, 2019.

[51] A. Krishnan, A. Sharma, A. Sankar, and H. Sundaram, "An adversarial approach to improve long-tail performance in neural collaborative filtering," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018, pp. 1491–1494.

[52] J. Estdale and E. Georgiadou, "Applying the iso/iec 25010 quality models to software product," in *Systems, Software and Services Process Improvement: 25th European Conference, EuroSPI 2018, Bilbao, Spain, September 5-7, 2018, Proceedings 25*. Springer, 2018, pp. 492–503.

[53] P. Botella, X. Burgués, J.-P. Carvallo, X. Franch, G. Grau, J. Marco, and C. Quer, "Iso/iec 9126 in practice: what do we need to know," in *Software Measurement European Forum*, vol. 2004, 2004.

[54] M. AZUMA, "Iso/iec draft cd 14598-1," 1995.

[55] F. Gualo, M. Rodríguez, J. Verdugo, I. Caballero, and M. Piattini, "Data quality certification using iso/iec 25012: Industrial experiences," *Journal of Systems and Software*, vol. 176, p. 110938, 2021.

[56] U. Aydan, M. Yilmaz, P. M. Clarke, and R. V. O'Connor, "Teaching iso/iec 12207 software lifecycle processes: a serious game approach," *Computer Standards & Interfaces*, vol. 54, pp. 129–138, 2017.

[57] E. Gökalp and V. Martinez, "Digital transformation capability maturity model enabling the assessment of industrial manufacturers," *Computers in Industry*, vol. 132, p. 103522, 2021.

[58] F. N. Colakoglu, A. Yazici, and A. Mishra, "Software product quality metrics: A systematic mapping study," *IEEE Access*, vol. 9, pp. 44 647–44 670, 2021.

[59] S. Pargaonkar, "A review of software quality models: A comprehensive analysis," *Journal of Science & Technology*, vol. 1, no. 1, pp. 40–53, 2020.

[60] F. Harper and J. Konstan, "The MovieLens datasets: History and context," *ACM Transactions on Interactive Intelligent Systems*, vol. 5, no. 4, pp. 1–19, 2015.

[61] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," in *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*, 2019, pp. 165–174.

[62] J. Ni, J. Li, and J. McAuley, "Justifying recommendations using distantly-labeled reviews and fined-grained aspects," in *Proc. EMNLP-IJCNLP*, 2019, pp. 188–197.

[63] J. McAuley, "Datasets for recommender systems research," https://cseweb.ucsd.edu/~jmcauley/datasets.html#amazon_reviews, n.d., accessed: 2024-09-12.

[64] L. Xia, C. Huang, Y. Xu, J. Zhao, D. Yin, and J. Huang, "Hypergraph contrastive collaborative filtering," in *Proceedings of the 45th International ACM SIGIR conference on research and development in information retrieval*, 2022, pp. 70–79.

[65] J. Yu, H. Yin, X. Xia, T. Chen, L. Cui, and Q. V. H. Nguyen, "Are graph augmentations necessary? simple graph contrastive learning for recommendation," in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2022, pp. 1294–1303.

[66] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proceedings of the 26th international conference on world wide web*, 2017, pp. 173–182.

[67] Z. Liu, L. Meng, F. Jiang, J. Zhang, and S. Y. Philip, "Deoscillated adaptive graph collaborative filtering," in *Topological, Algebraic and Geometric Learning Workshops 2022*. PMLR, 2022, pp. 248–257.

[68] X. Lin, W. Wang, J. Zhao, Y. Li, F. Feng, and T.-S. Chua, "Temporally and distributionally robust optimization for cold-start recommendation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 8, 2024, pp. 8750–8758.

[69] F. Huang, Z. Yang, J. Jiang, Y. Bei, Y. Zhang, and H. Chen, "Large language model interaction simulator for cold-start item recommendation," *arXiv preprint arXiv:2402.09176*, 2024.

[70] J. Monteil, V. Vaskovych, W. Lu, A. Majumder, and A. van den Hengel, "Marec: Metadata alignment for cold-start recommendation," in *Proceedings of the 18th ACM Conference on Recommender Systems*, 2024, pp. 401–410.

[71] S. Liu and Y. Zheng, "Long-tail session-based recommendation," in *Proceedings of the 14th ACM conference on recommender systems*, 2020, pp. 509–514.

[72] N. Achary and B. K. Patra, "Graph based hybrid approach for long-tail item recommendation in collaborative filtering," *ACM Digital Library*, 2020. [Online]. Available: https://dx.doi.org/10.1145/3430984.3431058

[73] Y. Wang, J. Wu, Z. Wu, and G. Adomavicius, "From head to long tail: Efficient and flexible recommendation using cosine patterns," in *From Head to Long Tail: Efficient and Flexible Recommendation Using Cosine Patterns: Wang, Yaqiong— uWu, Junjie— uWu, Zhiang— uAdomavicius, Gediminas*. [Sl]: SSRN, 2021.

[74] M. V. Sigova, I. K. Klioutchnikov, and O. I. Klioutchnikov, "Long-tail data-driven recommendations-innovative solutions for financial recommender systems," in *2021 International Conference Engineering Technologies and Computer Science (EnT)*. IEEE, 2021, pp. 92–97.

[75] R. Shafiloo, M. Kaedi, and A. Pourmiri, "Considering user dynamic preferences for mitigating negative effects of long tail in recommender systems," *arXiv preprint arXiv:2112.02406*, 2024. [Online]. Available: https://arxiv.org/abs/2112.02406

[76] G. Gupta and R. Katarya, "A novel approach to alleviate data sparsity and generate dynamic fruit recommendations from point-of-sale data," *Concurrency and Computation: Practice and Experience*, vol. 35, no. 1, p. e7423, 2023.

[77] K. S. Elattar and K. M. Fouad, "A survey on recommender systems challenges and solutions," in *2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)*. IEEE, 2022, pp. 296–301.

[78] I. Klioutchnikov and A. Klioutchnikova, "Long-tail distribution financial recommender systems embedded into social networks," *Communications of the IBIMA*, 2022. [Online]. Available: https://dx.doi.org/10.5171/2022.165496

[79] Z. Zhao, K. Zhou, X. Wang, W. X. Zhao, F. Pan, Z. Cao, and J.-R. Wen, "Alleviating the long-tail problem in conversational recommender systems," in *Proceedings of the 17th ACM Conference on Recommender Systems*, 2023, pp. 374–385.

[80] H. Yang, Y. Choi, G. Kim, and J.-H. Lee, "Loam: Improving long-tail session-based recommendation via niche walk augmentation and tail session mixup," in *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2023, pp. 527–536.

[81] M. Jangid and R. Kumar, "Deep learning approaches to address cold start and long tail challenges in recommendation systems: a systematic review," *Multimedia Tools and Applications*, pp. 1–33, 2024.

[82] J. Wu, C.-C. Chang, T. Yu, Z. He, J. Wang, Y. Hou, and J. McAuley, "Coral: Collaborative retrieval-augmented large language models improve long-tail recommendation," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024, pp. 3391–3401.

[83] Q. Liu, X. Wu, X. Zhao, Y. Wang, Z. Zhang, F. Tian, and Y. Zheng, "Large language models enhanced sequential recommendation for long-tail user and item," *arXiv preprint arXiv:2405.20646*, 2024.

[84] A. Gazdar and L. Hidri, "A new similarity measure for collaborative filtering based recommender systems," *Knowledge-Based Systems*, vol. 188, p. 105058, 2020.

[85] A. Pujahari and D. S. Sisodia, "Aggregation of preference relations to enhance the ranking quality of collaborative filtering based group recommender system," *Expert Systems with Applications*, vol. 156, p. 113476, 2020.

[86] F. Fkih, "Similarity measures for collaborative filtering-based recommender systems: Review and experimental comparison," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 9, pp. 7645–7669, 2022.

[87] J. Bobadilla, F. Ortega, A. Gutierrez, and A. Gonzalez-Prieto, "Deep variational models for collaborative filtering-based recommender systems," *Neural Computing and Applications*, vol. 35, no. 10, pp. 7817–7831, 2023.

[88] B. Alhijawi, N. Obeid, A. Awajan, and S. Tedmori, "New hybrid semantic-based collaborative filtering recommender systems," *International Journal of Information Technology*, vol. 14, no. 7, pp. 3449–3455, 2022.

[89] R. Widayanti, M. H. R. Chakim, C. Lukita, U. Rahardja, and N. Lutfiani, "Improving recommender systems using hybrid techniques of collaborative filtering and content-based filtering," *Journal of Applied Data Sciences*, vol. 4, no. 3, pp. 289–302, 2023.

[90] J. Noel, C. Monterola, and D. S. Tan, "Improving recommendation diversity without retraining from scratch," *International Journal of Data Science and Analytics*, pp. 1–10, 2024.

[91] A. Sharma, R. Raj, and B. Yamini, "Novel algorithmic recommendation engine for diverse content discovery," in *2024 11th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO)*. IEEE, 2024, pp. 1–4.

[92] M. Slokom and L. Hollink, "How to diversify any personalized recommender? a user-centric preprocessing approach," *arXiv preprint arXiv:2405.02156*, 2024.

[93] K. Zhao and J. Pi, "Research on rating error and quality metrics for collaborative filtering recommendation methods," in *Journal of Physics: Conference Series*, vol. 1302, no. 2. IOP Publishing, 2019, p. 022032.

[94] W. Juan, L. Yue-xin, and W. Chun-ying, "Survey of recommendation based on collaborative filtering," in *Journal of Physics: Conference Series*, vol. 1314, no. 1. IOP Publishing, 2019, p. 012078.

[95] B. Kumar, V. K. Nayak, P. Banerjee, and M. K. Dehury, "An empirical analysis of collaborative and content based recommender system by using ai," in *2023 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI)*. IEEE, 2023, pp. 1–8.

[96] S. Xiao, H. Ling, Y. Lu, and Z. Tang, "Study on collaborative filtering recommendation algorithm based on prediction for item rating," in *2020 International Conference on Information Science, Parallel and Distributed Systems (ISPDS)*. IEEE, 2020, pp. 352–355.

[97] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM computing surveys (CSUR)*, vol. 52, no. 1, pp. 1–38, 2019.

[98] P. Wang, "A collaborative filtering recommendation algorithm based on product clustering," *Applied Mechanics and Materials*, vol. 267, pp. 87–90, 2013.

[99] M. A. Ghazanfar and A. Prügel-Bennett, "An improved switching hybrid recommender system using naive bayes classifier and collaborative filtering," in *Proceedings of the International MultiConference of Engineers and Computer Scientists*, vol. 1, 2010.

[100] Y.-C. Hu, "A multicriteria collaborative filtering approach using the indifference relation and its application to initiator recommendation for group-buying," *Applied Artificial Intelligence*, vol. 28, no. 10, pp. 992–1008, 2014.

[101] M. Kuanr and P. Mohapatra, "Assessment methods for evaluation of recommender systems: a survey," *Foundations of Computing and Decision Sciences*, vol. 46, no. 4, pp. 393–421, 2021.

[102] V. W. Anelli, D. Malitesta, C. Pomo, A. Bellogín, E. Di Sciascio, and T. Di Noia, "Challenging the myth of graph collaborative filtering: a reasoned and reproducibility-driven analysis," in *Proceedings of the 17th ACM Conference on Recommender Systems*, 2023, pp. 350–361.

[103] W. Lei, X. He, M. de Rijke, and T.-S. Chua, "Conversational recommendation: Formulation, methods, and evaluation," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020, pp. 2425–2428.

[104] H. Abdollahpouri, M. Elahi, M. Mansoury, S. Sahebi, Z. Nazari, A. Chaney, and B. Loni, "Mors 2021: 1st workshop on multi-objective recommender systems," in *Proceedings of the 15th ACM Conference on Recommender Systems*, 2021, pp. 787–788.

[105] C. Huang, L. Xia, X. Wang, X. He, and D. Yin, "Self-supervised learning for recommendation," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022, pp. 5136–5139.

[106] J. Idakwo, J. B. Agbogun, and T. Kolajo, "A survey on recommendation system techniques," *UMYU Scientifica*, vol. 2, no. 2, pp. 112–119, 2023.

[107] G. V. Subhash *et al.*, "Optimizing recommender system: Literature review," *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 12, no. 10, pp. 3934–3939, 2021.

[108] D. R. Piyadigama and G. Poravi, "A review on pushing the limits of baseline recommendation systems with the integration of opinion mining & information retrieval techniques," *arXiv preprint arXiv:2205.01802*, 2022.

[109] Y. Zheng and D. Wang, "Multi-objective recommendations," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. ACM, Aug 2021, pp. 4098–4099.

[110] Y. Lin, Y. Liu, F. Lin, L. Zou, P. Wu, W. Zeng, H. Chen, and C. Miao, "A survey on reinforcement learning for recommender systems," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[111] Z. Zhang, T. Peng, and K. Shen, "Overview of collaborative filtering recommendation algorithms," in *IOP Conference Series: Earth and Environmental Science*, vol. 440, no. 2. IOP Publishing, 2020, p. 022063.

[112] L. Zhang, Z. Zhang, J. He, and Z. Zhang, "Ur: A user-based collaborative filtering recommendation system based on trust mechanism and time weighting," in *2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS)*. IEEE, 2019, pp. 69–76.

[113] X. Ren, L. Xia, Y. Yang, W. Wei, T. Wang, X. Cai, and C. Huang, "Sslrec: A self-supervised learning framework for recommendation," in *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, 2024, pp. 567–575.

[114] J. Deng, Q. Wu, S. Wang, J. Ye, P. Wang, and M. Du, "A novel joint neural collaborative filtering incorporating rating reliability," *Information Sciences*, vol. 665, p. 120406, 2024.

[115] K. Biliatdinov and V. Meniailo, "Modified method dea and methodology of technical systems effectiveness assessment," *Information technologies*, no. 11, pp. 611–617, 2020.

[116] E. Trichkova-Kashamova, "Applying the iso/iec 25010 quality models to an assessment approach for information systems," in *2021 12th National Conference with International Participation (ELECTRONICA)*. IEEE, 2021, pp. 1–4.

[117] M. I. F. Nuzula and S. Rochimah, "Evaluation of service quality in human resource information systems using the iso/iec 25010," in *2023 International Seminar on Application for Technology of Information and Communication (iSemantic)*. IEEE, 2023, pp. 215–220.

[118] W. Al Jurdi, J. B. Abdo, J. Demerjian, and A. Makhoul, "Group validation in recommender systems: Framework for multi-layer performance evaluation," *ACM Transactions on Recommender Systems*, vol. 2, no. 1, pp. 1–25, 2024.

[119] A. Jadon and A. Patil, "A comprehensive survey of evaluation techniques for recommendation systems," in *International Conference on Computation of Artificial Intelligence & Machine Learning*. Springer, 2024, pp. 281–304.

[120] M. D. Ekstrand, B. Carterette, and F. Diaz, "Distributionally-informed recommender system evaluation," *ACM Transactions on Recommender Systems*, vol. 2, no. 1, pp. 1–27, 2024.

[121] O. A. Omitaomu, H. B. Klasky, M. Olama, O. Ozmen, L. Pullum, A. M. Thakur, T. Kuruganti, J. M. Scott, A. Laurio, F. Drews *et al.*, "A new methodological framework for hazard detection models in health information technology systems," *Journal of biomedical informatics*, vol. 124, p. 103937, 2021.

[122] M. Mali, D. S. Mishra, and M. Vijayalaxmi, "Multifaceted recommender systems methods: A review," *Journal of Statistics and Management Systems*, vol. 23, no. 2, pp. 349–361, 2020.

[123] C. Grant and A. Osanloo, "Understanding, selecting, and integrating a theoretical framework in dissertation research: Creating the blueprint for your "house"," *Administrative issues journal*, vol. 4, no. 2, p. 4, 2014.

[124] G. Blanco and A. Lourenço, "A multilayered graph-based framework to explore behavioural phenomena in social media conversations," *International Journal of Medical Informatics*, vol. 179, p. 105236, 2023.

[125] J. Mao, Y. Zhu, M. Chen, G. Chen, C. Yan, and D. Liu, "A contradiction solving method for complex product conceptual design based on deep learning and technological evolution patterns," *Advanced Engineering Informatics*, vol. 55, p. 101825, 2023.

[126] D. M. Buede and W. D. Miller, *The engineering design of systems: models and methods*. John Wiley & Sons, 2024.

[127] INCOSE, *INCOSE systems engineering handbook*. John Wiley & Sons, 2023.

[128] N. Phadnis and M. Torkkeli, "Impact of theory of inventive problem solving (triz) on innovation portfolio development," in *International TRIZ Future Conference*. Springer, 2023, pp. 342–359.

[129] K. Haryono, I. A. N. Fathony, and R. C. Nugraha, "A proposed prototype of triz mobile application in business and management," in *AIP Conference Proceedings*, vol. 2828, no. 1. AIP Publishing, 2023.

[130] B. Yıldırım, "Flipped triz-stem: Enhancing teacher training through innovative pedagogy?" *Education and Information Technologies*, vol. 29, no. 9, pp. 10 899–10 929, 2024.

[131] H. Kuwajima, H. Yasuoka, and T. Nakae, "Engineering problems in machine learning systems," *Machine Learning*, vol. 109, no. 5, pp. 1103–1126, 2020.

[132] J. Han, K. S. Park, and K. M. Lee, "An automated machine learning platform for non-experts," in *Proceedings of the International Conference on Research in Adaptive and Convergent Systems*, 2020, pp. 84–86.

[133] F. F. Prado and L. A. Digiampietri, "A systematic review of automated feature engineering solutions in machine learning problems," in *Proceedings of the XVI Brazilian Symposium on Information Systems*, 2020, pp. 1–7.

[134] S. Mohseni, "Toward design and evaluation framework for interpretable machine learning systems," in *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, 2019, pp. 553–554.

[135] U. Kannengiesser and J. S. Gero, "Modelling the design of models: an example using crisp-dm," *Proceedings of the Design Society*, vol. 3, pp. 2705–2714, 2023.

[136] S. Moreschini, F. Lomio, D. Hästbacka, and D. Taibi, "Mlops for evolvable ai intensive software systems," in *2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*. IEEE, 2022, pp. 1293–1294.

[137] A. S. Jakobsen, "Study of dataops as a concept for aker bp to enable data-driven assets." Master's thesis, uis, 2022.

[138] W. Hummer, V. Muthusamy, T. Rausch, P. Dube, K. El Maghraoui, A. Murthi, and P. Oum, "Modelops: Cloud-based lifecycle management for reliable and trusted ai," in *2019 IEEE International Conference on Cloud Engineering (IC2E)*. IEEE, 2019, pp. 113–120.

[139] S. Chenoweth and P. K. Linos, "Teaching machine learning as part of agile software engineering," *IEEE Transactions on Education*, 2023.

[140] R. Henriques and L. Pinto, "A novel evaluation framework for recommender systems in big data environments," *Expert Systems With Applications*, vol. 231, p. 120659, 2023.

[141] J. Wang, Y. Huang, C. Chen, Z. Liu, S. Wang, and Q. Wang, "Software testing with large language models: Survey, landscape, and vision," *IEEE Transactions on Software Engineering*, 2024.

[142] L. He, X. Wang, D. Wang, H. Zou, H. Yin, and G. Xu, "Simplifying graph-based collaborative filtering for recommendation," in *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, 2023, pp. 60–68.

[143] W. X. Zhao, S. Mu, Y. Hou, Z. Lin, Y. Chen, X. Pan, K. Li, Y. Lu, H. Wang, C. Tian, Y. Min, Z. Feng, X. Fan, X. Chen, P. Wang, W. Ji, Y. Li, X. Wang, and J. Wen, "Recbole: Towards a unified, comprehensive and efficient framework for recommendation algorithms," in *CIKM*. ACM, 2021, pp. 4653–4664.

[144] W. X. Zhao, Y. Hou, X. Pan, C. Yang, Z. Zhang, Z. Lin, J. Zhang, S. Bian, J. Tang, W. Sun, Y. Chen, L. Xu, G. Zhang, Z. Tian, C. Tian, S. Mu, X. Fan, X. Chen, and J. Wen, "Recbole 2.0: Towards a more up-to-date recommendation library," in *CIKM*. ACM, 2022, pp. 4722–4726.

[145] L. Xu, Z. Tian, G. Zhang, J. Zhang, L. Wang, B. Zheng, Y. Li, J. Tang, Z. Zhang, Y. Hou *et al.*, "Towards a more user-friendly and easy-to-use benchmark library for recommender systems," in *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2023, pp. 2837–2847.

[146] RUCAIBox, "Recbole: A unified, comprehensive and efficient recommendation library," https://github.com/RUCAIBox/RecBole, 2024, accessed: 2024-04-28.

[147] A. Ortega, P. Frossard, J. Kovačević, J. M. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 808–828, 2018.

[148] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[149] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," *Advances in neural information processing systems*, vol. 29, 2016.

[150] K. Xu, C. Li, Y. Tian, T. Sonobe, K.-i. Kawarabayashi, and S. Jegelka, "Representation learning on graphs with jumping knowledge networks," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5453–5462.

[151] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, and K. Weinberger, "Simplifying graph convolutional networks," in *International conference on machine learning*. PMLR, 2019, pp. 6861–6871.

[152] L. Datta, "A survey on activation functions and their relation with xavier and he normal initialization," *arXiv preprint arXiv:2004.06632*, 2020.

[153] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[154] J. M. Jianmo Ni, Jiacheng Li. (2018) Amazon review data (2018). Accessed June 11, 2023. [Online]. Available: https://nijianmo.github.io/amazon/index.html

[155] C. Donnat, M. Zitnik, D. Hallac, and J. Leskovec, "Learning structural node embeddings via diffusion wavelets," in *Proc. ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 1320–1329.

[156] B. Xu, H. Shen, Q. Cao, K. Cen, and X. Cheng, "Graph convolutional networks using heat kernel for semi-supervised learning," *arXiv preprint arXiv:2007.16002*, 2020.

[157] M. Ahmed, R. Seraj, and S. M. S. Islam, "The k-means algorithm: A comprehensive survey and performance evaluation," *Electronics*, vol. 9, no. 8, p. 1295, 2020.

[158] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.

[159] J. G. Proakis, *Digital signal processing: principles, algorithms, and applications, 4/E*. Pearson Education India, 2007.

[160] F. Xue, X. He, X. Wang, J. Xu, K. Liu, and R. Hong, "Deep item-based collaborative filtering for top-n recommendation," *ACM Transactions on Information Systems (TOIS)*, vol. 37, no. 3, pp. 1–25, 2019.

[161] D.-K. Chae, J.-S. Kang, S.-W. Kim, and J.-T. Lee, "CFGAN: A generic collaborative filtering framework based on generative adversarial networks," in *Proc. ACM international Conference on Information and Knowledge Management*, 2018, pp. 137–146.

[162] R. v. d. Berg, T. N. Kipf, and M. Welling, "Graph convolutional matrix completion," *arXiv preprint arXiv:1706.02263*, 2017.

[163] N. Zhang, X. Guan, J. Cao, X. Wang, and H. Wu, "A hybrid traffic speed forecasting approach integrating wavelet transform and motif-based graph convolutional recurrent neural network," *arXiv preprint arXiv:1904.06656*, 2019.

[164] S. Boutemedjet and D. Ziou, "A graphical model for context-aware visual content recommendation," *IEEE Transactions on Multimedia*, vol. 10, no. 1, pp. 52–62, 2008.

[165] J. Konstan, B. Miller, D. Maltz, J. Herlocker, L. Gordon, and J. Riedl, "Grouplens: Applying collaborative filtering to usenet news," *Communications of the ACM*, vol. 40, no. 3, pp. 77–87, 1997.

[166] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," *2008 Eighth IEEE International Conference on Data Mining*, pp. 263–272, 2008.

[167] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[168] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," in *Proc. Conference on Uncertainty in Artificial Intelligence*, 2009, pp. 452–461.

[169] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2008, pp. 426–434.

[170] D. Liang, R. G. Krishnan, M. D. Hoffman, and T. Jebara, "Variational autoencoders for collaborative filtering," in *Proc. International Conference on World Wide Web*, 2018, pp. 689–698.

[171] X. Wang, H. Jin, A. Zhang, X. He, T. Xu, and T.-S. Chua, "Disentangled graph collaborative filtering," in *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, 2020, pp. 1001–1010.

[172] J. Chang, C. Gao, X. He, D. Jin, and Y. Li, "Bundle recommendation with graph convolutional networks," in *Proceedings of the 43rd international ACM SIGIR conference on Research and development in Information Retrieval*, 2020, pp. 1673–1676.

[173] Z. Li, Q. Xu, Y. Jiang, K. Ma, X. Cao, and Q. Huang, "Neural collaborative preference learning with pairwise comparisons," *IEEE Transactions on Multimedia*, vol. 23, pp. 1977–1989, 2020.

[174] J. Jin, J. Qin, Y. Fang, K. Du, W. Zhang, Y. Yu, Z. Zhang, and A. J. Smola, "An efficient neighborhood-based interaction model for recommendation on heterogeneous graph," in *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 2020, pp. 75–84.

[175] L. Chen, L. Wu, R. Hong, K. Zhang, and M. Wang, "Revisiting graph based collaborative filtering: A linear residual graph convolutional network approach," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 01, 2020, pp. 27–34.

[176] S. Sedhain, A. K. Menon, S. Sanner, and L. Xie, "Autorec: Autoencoders meet collaborative filtering," in *Proceedings of the 24th international conference on World Wide Web*, 2015, pp. 111–112.

[177] H. Wang, N. Wang, and D.-Y. Yeung, "Collaborative deep learning for recommender systems," in *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 2015, pp. 1235–1244.

[178] H. Zhang and J. McAuley, "Stacked mixed-order graph convolutional networks for collaborative filtering," in *Proc. SIAM International Conference on Data Mining*, 2020, pp. 73–81.

[179] C. Godsil and G. Royle, *Algebraic Graph Theory*. Springer-Verlag New York, 2001.

[180] H. Krim and A. Ben Hamza, *Geometric Methods in Signal and Image Analysis*. Cambridge University Press, 2015.

[181] D. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 129–150, 2011.

[182] C. Li and A. Ben Hamza, "A multiresolution descriptor for deformable 3D shape retrieval," *The Visual Computer*, vol. 29, pp. 513–524, 2013.

[183] I. Yeo and R. Johnson, "A new family of power transformations to improve normality or symmetry," *Biometrika*, vol. 87, pp. 954–959, 2000.

.1 Appendix 1: Additional Material

Each Solution in this table reflects a chapter in the thesis. Moreover, each solution is formed of techniques and each technique has a specific effect on the challenges presented in Section 1.4.

ISO 25010 Evaluation in Solution 1 addresses key challenges in Section 1.4 effectively. CPM in Solution 1 (TGCF) and Solution 2 (LSGCN) effectively handle multiple challenges but have some reproducibility issues. Simplified GCN and Adaptive GWT Techniques both attempt to handle some challenges but provide a supportive role essentially. Some of the techniques have a broad impact and can be generalized, while some have supportive roles, illustrating the need for a multi-faceted approach to coordinate handling CFRS challenges.

Table 1: Assessment of The Proposed Solutions and Their Techniques for Addressing Various CFRS Challenges.

Solution	Technique	Diversity & Novelty	Fairness	Long-Tail	Cold Start	Ranking & Relevancy	Reproducibility	Overfitting
A Methodological Framework	SBD	C	C	C	C	C	P	C
	FA	C	C	C	C	C	P	C
	TRIZ	C	C	C	C	C	P	C
	TRIZ-DSS	C	C	C	C	C	P	C
Solution1: TGCF	TFI	P	C	C	C	P	N	P
	CPM	P	P	P	P	C	N	P
	ISO 25010 Evaluation	P	P	P	P	P	P	P
	Shannon Entropy	P	C	P	C	N	C	C
	Recall Experiment	N	N	N	N	P	N	N
	NDCG Experiment	N	N	N	N	P	N	N
	Gini Index Experiment	C	P	C	C	N	C	C
	Popularity Experiment	C	C	C	C	C	C	P
	Item Coverage Experiment	P	P	P	P	N	C	P
	Loss Metrics	N	N	N	N	P	C	N
Solution 2: LSGCN	CPM	P	P	P	P	C	N	P
	Simplified GCN	C	C	C	C	C	C	N
	NDCG@k Experiment	N	N	N	N	P	N	N
	Coverage@k Experiment	P	P	P	P	N	C	P
	GiniIndex@k Experiment	C	P	C	C	N	C	C
	Epochs Increment	N	N	N	N	P	C	C
	Varying Density Groups	C	C	P	P	P	C	C
	Statistical Significance	N	N	N	N	P	P	N
Solution 3: SAWE	Adaptive GWT	C	C	C	C	C	N	C
	Cluster-Based Filtering	C	C	C	C	C	N	P
	Varying Density Groups	C	C	P	P	P	C	C
	Sparse Dataset Experiment	C	C	C	P	C	C	C
	NDCG@k Multilayer	N	N	N	N	P	N	N
	Epochs Increment	N	N	N	N	P	C	C
Solution 3: CAGLE	Adaptive GWT	C	C	C	C	C	N	C
	ATF	C	C	C	C	C	N	C
	NDCG@k Experiment	N	N	N	N	P	N	N
	Sparse Dataset Experiment	C	C	C	P	C	C	C
	Varying Density Groups Experiment	C	C	P	P	P	C	C
	Epochs Increment	N	N	N	N	P	C	C

- **P: Problem Addressed** - The technique directly impacts and addresses the challenge.
- **C: Problem Mitigated** - The technique partially impacts and mitigates the challenge.
- **N: No Direct Impact** - The technique does not significantly impact the challenge.

.2 Appendix 2: Additional Material

The table below reorganizes the practical criteria that define high, medium, and low scores across several dimensions, [1,7,97] which also help in understanding the capabilities and limitations across different systems based on personalization, data sparsity, cold start, and scalability features.

Table 2: Practical Criteria for Defining High, Medium, Low Rankings

Criteria	High	Medium	Low
Personalization	- Making great use of user-specific data. - Utilizes advanced algorithms (e.g., collaborative filtering). - Very personalized recommendations. - Examples: Netflix, Amazon.	- Moderate use of user data. - Mixture of general and specific suggestions. - Fair balanced approach. - Examples: News aggregators.	- Minimal use of user data. - Generic recommendations based on broad segments. - Examples: Traditional retail websites.
Data Sparsity	- Performs well with limited data. - Strategies to infer missing data (e.g., matrix factorization). - Robust to sparse interactions. - Examples: Spotify's collaborative filtering.	- Can handle medium data sparsity but requires moderate data. - Uses additional data sources to fill gaps. - Examples: Hybrid models combining CF and content-based filtering.	- Struggles with sparse data. - Requires dense datasets. - Examples: Basic collaborative filtering without advanced techniques.
Cold Start	- Major challenge with new users/items. - Needs historical data for good performance. - Examples: Traditional collaborative filtering.	- Moderate difficulty, uses additional data (e.g., demographic, content-based) to alleviate issues. - Examples: Hybrid systems.	- Minimal difficulty. - Effectively uses external data (e.g., item features, domain knowledge). - Examples: Content-based systems, knowledge-based systems.
Scalability	- Maintains performance with very large datasets. - Efficient algorithms and infrastructure. - Examples: Large-scale recommender systems like Amazon.	- Good scalability but potential performance issues with very large datasets. - Examples: Mid-sized E-commerce platforms.	- Low scalability. - Significant performance degradation as data size increases. - Examples: Small-scale bespoke systems.

[1] Examples reflect practical applications for each ranking level.