***Accuracy and Prediction Performance of Collaborative Filtering Algorithms Using Deep Learning.***



**A Thesis**

**By**

**ALAMGIR HOSSAIN**

**ID: G21275002**

**Supervisor: *Dr Panayiotis Andreou***

**Deputy Head Of The School Of Sciences, Associate Professor In Data Management And Course Leader Of Msc Data Analytics**

**Email:PGAndreou@uclan.ac.uk**

**Submitted in Partial Fulfilment of the requirements of**

**University of Central Lancashire**

**for the degree of**

**Master of Science in Data Analytics**

**18 July, 2025**

**Declaration**

This dissertation is the result of my individual efforts and does not include any work done in collaboration, except where specifically noted in the text. It has not been submitted, either in part or in whole, to any university or institution for any degree, diploma, or other qualification.

I have reviewed the information regarding plagiarism as outlined in the University Academic Regulations and the Assessment Handbook.

Name of Student Date

Alamgir Hossain 18.07.2025

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

University of Central Lancashire, Cyprus

**Abstract**

Recommendation systems have grown to be an inseparable part of newer digital platforms, and they are now forming product discovery, content, and relationship finding for users. One of the most popular methods is Collaborative Filtering (CF), which is based on past interactions of users and items to predict user preferences. Nonetheless, CF also has two significant shortcomings: data sparsity - most users only interact with a few items - and item synonymy; semantically identical items are not considered synonymous, because different words represent them. These problems prevent the system from generating effective and varied recommendations, especially in cold-start settings or when handling semantically intensive queries. This research paper introduces ConCF (Collaborative Neural Content Fusion), a novel deep learning framework that combines Neural Collaborative Filtering (NCF) with Convolutional Neural Networks (CNN) to address these challenges. Although NCF aims to model latent user-item relationships using embeddings, it cannot utilize semantic data found in content. CNNs are, however, good at deriving local semantic attributes from textual metadata. ConCF combines the two architectures and therefore represents both structured-behavioral data and unstructured material, encapsulating a one-stop, accumulated answer to issues of sparsity and synonymy. The MovieLens 1M dataset tests the model in different sparsity conditions. ConCF outperforms four baseline models (Autoencoder, NCF, ConvMF, and Supervised ConvMF) with a minimum RMSE of 0.877 and a highest Recall@5 of 53.70% at 20% sparsity, providing evidence of its strength and generalizability. These results show that ConCF is a scalable, semantically-aware recommendation solution that can be deployed in practice.

**Table of Contents**

**Page**

DECLARATION.........................................................................................................................................II

ABSTRACT..............................................................................................................................................III

TABLE OF CONTENTS ............................................................................................................................IV

LIST OF FIGURES ..................................................................................................................................VII

LIST OF TABLES....................................................................................................................................VIII

ACKNOWLEDGEMENTS .........................................................................................................................IX

1. **INTRODUCTION** ... .............................................................................................................................1

1.1 RESEARCH GAP AND MOTIVATION.............................................................................................1

1.2 RESEARCH OBJECTIVES AND CONTRIBUTIONS....................................................................... ….3

2. **LITERATURE REVIEW**..........................................................................................................................4

2.1 INTRODUCTION TO COLLABORATIVE FILTERING.........................................................................4

2.2 MATRIX FACTORIZATION AND LINKED DATA…………….……..........................................................5

2.3 REVIEW-BASED AND HYBRID FILTERING TECHNIQUES................................................................6

2.4 DEEP LEARNING IN RECOMMENDER SYSTEMS...........................................................................6

2.5. HYBRID DEEP LEARNING MODELS..…………………….................................................................. ….7

2.6 CNN-BASED CF FOR SPARSITY AND SYNONYMY..........................................................................7

3. **METHODOLOGY**.................................................................................................................................8

3.1 RESEARCH DESIGN AND FRAMEWORK........................................................................................8

3.1.1 METHODOLOGY FLOW DIAGRAM……………….....................................................................9

3.2 DATASET DESCRIPTION AND EXPLORATORY DATA ANALYSIS....................................................10

3.2.1 DATASET DESCRIPTION………………………………………………………………………………………………..10

3.2.2 EXPLORATORY DATA ANALYSIS (EDA)………………………………………………………………………….10

3.3 DATA PREPROCESSING.............................................................................................................16

3.3.1 PREPROCESSING FOR AUTOENCODER.............................................................................17

3.3.2 PREPROCESSING FOR NCF, CONVMF, SUPERVISED CONVMF, AND CONCF………………….17

3.3.3 COMPARISON WITH SUPERVISED CONVMF PREPROCESSING………………………………………18

3.3.4 SUMMARY OF PREPROCESSING PIPELINES…………………………………………………………………..19

3.4 BASELINE MODELS……………………………………………………………………………………………………………..20

3.4.1 AUTOENCODER (AE)………………………………………………………………………………………………….20

3.4.2 NEURAL COLLABORATIVE FILTERING (NCF)……………………………………………………………….21

3.5 PROPOSED HYBRID MODEL: CONCF (COLLABORATIVE NEURAL CONTENT FUSION)…………….22

3.5.1 MOTIVATION AND DESIGN RATIONALE…………………………………………………………………….23

3.5.2 MODEL ARCHITECTURE……………………………………………………………………………………………..23

3.5.3 MATHEMATICAL FORMULATION AND PREDICTION CALCULATION……………………………24

3.5.4 MODEL DIAGRAM……………………………………………………………………………………………………..25

3.5.5 IMPLEMENTATION SUMMARY…………………………………………………………………………………25

3.5.6 HYPERPARAMETER CONFIGURATION AND TUNING STRATEGY…………………………………26

3.5.7 EVALUATION SETUP…………………………………………………………………………………………………29

3.5.7.1 JUSTIFICATION FOR 20% SPARSITY BENCHMARK…………………………………………..29

3.5.8 MODEL LIMITATIONS AND DESIGN TRADE-OFFS………………………………………………………..30

3.5.8.1 DESIGN TRADE-OFFS: WORD EMBEDDINGS AND MODEL ARCHITECTURE………30

3.5.9 TRANSFERABILITY OF RESULTS…………………………………………………………………………………..31

3.5.10 MODEL COMPARISON OVERVIEW………………………………………………………………………......32

4. **RESULTS**...........................................................................................................................................33

4.1 EVALUATING SPARSITY EFFECTS: AUTOENCODER VS. NCF................................ ………………..33

4.2 COMPARING CONCF WITH BASELINE MODELS………………….................................................34

4.2.1 COMPARATIVE INSIGHTS..................................................……………………………………….35

4.3. MODEL VALIDATION UNDER 20% SPARSITY..…………………………………………………………………..36

4.4 EVALUATION OF RECOMMENDATION QUALITY...................................................................38

4.4.1 REGRESSION-BASED EVALUATION..............................................................................39

4.4.2 RANKING-BASED TOP-N RECOMMENDATION EVALUATION......................................39

5. **DISCUSSION**.....................................................................................................................................40

5.1 SUMMARY OF KEY RESULTS.................................................................................................41

5.2 INTERPRETING THE IMPROVEMENTS OVER BASELINE........................................................41

5.3 IMPLICATIONS FOR REAL-WORLD RECOMMENDER SYSTEMS.............................................41

5.4 THEORETICAL CONTRIBUTION..............................................................................................42

5.5 LIMITATIONS AND DESIGN TRADE-OFFS…………………………………………………………………………..42

5.6 SUMMARY............................................................................................................................42

5.7 ETHICAL AND LEGAL CONSIDERATIONS...............................................................................42

6. **CONCLUSION AND FUTURE WORK**..................................................................................................43

6.1 CONCLUSION............................................................................................................................43

6.2 FUTURE RESEARCH DIRECTION................................................................................................44

6.3 FINAL REMARKS…………………………………………………………………………………………………………………..45

**List of Figures**

**Figure Page**

**3.1 Schematic of a suggested methodology pipeline, which demonstrates the preparation of data, the design of the models (Autoencoder, NCF, ConvMF, Super-ConvMF, and ConCF), and their training on sparse inputs, and evaluation based on performance measures..................................................................................................................................................9**

**3.2 Ratings Distribution.......................................................................................................................11**

**3.3 The top-rated movies (all with an average rating of 5.0) include..............................................11**

**3.4 User and Items Rating Distribution.............................................................................................12**

**3.5 Genre Distribution.........................................................................................................................13**

**3.6 Sparse User-Item ratings Matrix..................................................................................................14**

**3.7 User Age Group Distribution........................................................................................................15**

**3.8 User Age Group Distribution........................................................................................................15**

**3.9 Top 10 User Occupations..............................................................................................................16**

**3.10 Architecture of the proposed ConCF model combining user/movie embeddings and CNN-based semantic content representation..............................................................................................25**

**4.1 RMSE vs. Sparseness: Autoencoder vs. NCF..............................................................................34**

**4.2 RMSE vs. Sparsity for all models................................................................................................,35**

**4.3 Loss Curve of ConCF model.........................................................................................................36**

**4.4 Actual vs. Predicted Ratings.........................................................................................................37**

**4.5 Top-5 Recommendation output....................................................................................................38**

**List of Tables**

**TABLE PAGE**

**3.1 Comparison with Supervised ConvMF Preprocessing Model……………………………………19**

**3.2 Summary of Preprocessing Pipelines……………………………………………………………………………….19**

**3.3 Baseline Model Comparison Overview (as reported in Ref. Liu et al., (2018, p. 1850018-13…………………………………………………………………………………………………………………………………………22**

**3.4 Symbol Description…………………………………………………………………………………………………………25**

**3.5 Model Architecture Hyperparameters………………………………………………..............................27**

**3.6 Training Configuration and Dynamic Learning…………………………………...............................28**

**3.7 Comparison of Baseline and Proposed Models……………………………………………………33**

**4.1 Indicative Benchmarking of Top-N Recommendation Metrics (Tasks Differ – Not a Direct Baseline Evaluation)………………………………………………………………………………………40**

**Acknowledgements**

*All praise and thanks are due to Almighty Allah, the Creator of the world, whose grace and blessings have enabled me to complete this research. I am deeply grateful to everyone who has provided support, encouragement, and guidance throughout my academic journey and the completion of this thesis. First and foremost, I wish to express my heartfelt gratitude to my beloved parents. Their unwavering support, encouragement, and inspiration have been instrumental in helping me reach this stage of my academic career. Their belief in my potential has consistently motivated me. I would like to extend my sincere appreciation to my esteemed supervisor,* **Dr. Panayiotis Andreou***,* **Deputy Head Of The School of Sciences, Associate Professor in Data Management, and Course Leader of the MSc Data Analytics** *at UCLAN, Cyprus. His continuous guidance, insightful feedback, and valuable advice have been pivotal to my academic progress. His regular communication and support were crucial to the successful completion of this research. I am also profoundly grateful to* ***Dr Christiana Ioannou****,* ***Assistant Professor in Cybersecurity, Course Leader of MSc Cybersecurity / MSc Cybersecurity Distance Learning****, UCLAN, Cyprus, for his invaluable support and guidance in achieving my whole research knowledge from the beginning through Critical Analysis and Master’s Projects Classes. I extend my heartfelt thanks to all the faculty members of* ***the Data Analytics program at the University of Central Lancashire, Cyprus****, for providing a stimulating academic environment and access to the resources necessary for conducting this research. Additionally, I would like to thank the honorable members of the internal and external assessment board for their time, effort, and constructive feedback in reviewing my thesis.*

*Lastly, I acknowledge all those who directly or indirectly contributed to the successful completion of this thesis. Your support will always be remembered.*

*With sincere gratitude,*

*Alamgir Hossain*

**Chapter 1: Introduction**

1. **Research Gap and Motivation**

Recommender systems have become essential components of digital ecosystems, fundamentally transforming the way users interact with online content, services, and platforms. Whether it is the personalized shopping recommendations on Amazon, the personalized music playlists on Spotify, or the movie-wise recommendations on Netflix, these systems are central to an increase in user experience and satisfaction (Strömqvist, 2018; Xiao et al., 2018). In their beginnings, recommender systems seek to expose relevant and in many cases serendipitous material through comparison of trends in user behaviour, user preferences, and user history, to provide them then (Cai et al., 2020, p. 668).

Collaborative Filtering (CF) system is one of the most well-known paradigms in recommendation systems, as the latent similarity information in user-item interaction data like ratings, clicks, or purchase behavior is extracted to infer the preferences of users. There are two types of CF: user-based and item-based approaches (Ricci et al., 2010; Schafer et al., 2007). The user-based methods suggest products that other people who are similar to the user have liked, whereas the item-based methods are concerned with what is more likely to be liked with another item. Although they demonstrate high levels of scalability and simplicity, traditional forms of CF have drawbacks, wherein two, the most common challenges emerge in the real world (Moe & Htwe, 2017; Natarajan et al., 2020):

1. **Data Sparsity:** Most users interact with only a small portion of the available item catalog, which makes the user-item matrix quite sparse. This sparsity limits the system's ability to accurately predict user preferences, particularly when it encounters new users or new items.
2. **Item Synonymy:** Collaborative Filtering (CF) models treat items as distinct entities, failing to recognize the semantic similarity between them. As a result, two movies categorized as action films but with different titles may not be considered substitutes, which restricts diversity and authenticity in recommendations.

With the rapid growth of digital content and the increasing number of users, there is a growing need for efficient and intelligent recommendation systems. However, these systems often fail due to two main issues: the incomplete interaction histories and the inability to identify semantically equivalent items. These shortcomings not only limit the effectiveness of recommendations but also hinder user trust and satisfaction.

Data sparsity has consistently been a significant challenge for the success of recommendation systems. On platforms with thousands or even millions of items, users typically explore only a small fraction of them, based on the data available. This results in matrices that contain a large percentage of missing entries. Additionally, item synonymy—where different items serve the same user intention—creates semantic ambiguity that cannot be resolved using traditional collaborative filtering (CF) methods, which rely on discrete item identifiers.

Significant advances have been made in the field of collaborative filtering (CF), yet there remains a considerable research gap in addressing data sparsity and item synonymy within a unified non-supervised architecture. Traditional models, such as Matrix Factorization (MF) and Autoencoders, focus solely on observed interactions between users and items. While these models perform well with dense datasets, their effectiveness diminishes when user interactions are limited, which is often the case in real-world situations.

Neural Collaborative Filtering (NCF) has improved upon classical CF by capturing user-item interactions in a non-linear manner using deep learning techniques. However, NCF still struggles to incorporate item content or recognize semantically similar items. This limitation hampers its ability to effectively tackle the problem of synonymy, where semantically related items (such as two films sharing similar themes or genres) may not be treated as interchangeable.

To bridge this gap, researchers have explored content-aware models that utilize item metadata—typically in the form of tags or descriptive text—through techniques like Convolutional Matrix Factorization (ConvMF). Nevertheless, these models often require supervision (e.g., annotated tags), which can be impractical to obtain at scale and may hinder their generalizability. Furthermore, many recent models have not explored joint architectures that:

1. Function efficiently in highly sparse environments,
2. Extract textual features from the context of item characteristics, and
3. Continuously learn without the reliance on annotated or labeled data.

In light of such issues, contemporary studies have focused on achieving hybrid architectures, where collaborative signals and item content interact. The Neural Collaborative Filtering (NCF) framework has shown promise in modeling both implicit and explicit interactions, particularly through the application of nonlinear layers. However, as indicated, NCF does not use text or semantic information. As an alternative, Convolutional Neural Networks (CNNs) have proven effective in learning semantic text information, which is helpful in applications such as sentiment analysis, document classification, and content-based recommendation. These advantages hint at a potential convergence: a hybrid of the synergy between the collaborative strength of NCF and the semantic knowledge of CNN can yield an even stronger organization.

This thesis has been provoked by the willingness to create such a model. It proposes ConCF (Collaborative Neural Content Fusion), a new architecture that combines the concept of deep collaborative filtering (through NCF) with semantic feature extraction (through CNNs). The model is trained using both user-item ratings and free-form movie metadata (including movie names and genre names), in a fully unsupervised fashion. Analyzing the performance on a multi-tier scale of data sparsity, the study demonstrates that ConCF can provide semantically sensitive, scalable, and robust recommendations, satisfying both real-life and academic requirements for recommender systems.

## ****Research Objectives and Contributions****

The purpose of this research is to create and rigorously test a hybrid deep learning model that addresses two major drawbacks of collaborative filtering: data sparsity and item synonymy. To evaluate the proposed Collaborative Filtering Model (ConCF), we will use two datasets, MovieLens 1M and MovieLens 20M. It is recommended that future studies replicate the proposed model with the same datasets and experimental settings to ensure consistency and applicability.

1. **Objectives:**
2. To identify the shortcomings of classical and deep collaborative filtering models in sparse and semantically ambiguous environments.
3. To propose ConCF, a hybrid architecture that combines:
   1. Neural Collaborative Filtering (NCF) to model latent interactions between users and items.
   2. Convolutional Neural Network (CNN) to generate semantic representations of item metadata, such as title and genre.
4. To compare the performance of the model with varying training densities (20%, 40%, 60%, and 80%), focusing on the accuracy of rating predictions and the relevance of recommendations.

### To benchmark ConCF against popular models, including Autoencoder, NCF, ConvMF, and Supervised ConvMF, under the same data conditions.

### ****Key Contributions****

1. This study presents ConCF, a new model that hybridizes Neural Collaborative Filtering (NCF) and Convolutional Neural Network (CNN) components within a deep learning framework to enhance collaborative filtering.
2. The results indicate improved performance, with all high-sparsity scenarios showing the lowest Root Mean Square Error (RMSE) of 0.877 and the highest Recall@5 of 53.70% compared to baseline models. Notably, ConCF is the only model that outperforms all baseline results across all scenarios.
3. The use of pretrained GloVe embeddings combined with CNN layers facilitates the integration of semantic content, addressing item synonymy that would typically require additional supervision.
4. A standardized, replicable, and examinable experimental design was implemented to allow reliable comparisons across varying degrees of sparsity and among different models.

These contributions shed light on the limitations of the current state-of-the-art in recommender systems. The study bridges the gap between behavioral data and content semantics, proposing a generalizable, content-aware, and high-performance hybrid model.

**Chapter 2: Literature Review**

This review chapter discusses collaborative filtering, content-based methods, and hybrid deep learning approaches, focusing on the enduring challenges of data sparsity and item synonymy that affect real-world recommendation systems.

1. **Introduction to Collaborative Filtering**

Collaborative Filtering (CF) is a fundamental methodology in recommender systems, and its functioning is based on historical data about user and item interactions, such as ratings, purchases, clicks, or views, to determine user preferences and make recommendations. Its logic is based on the statement that past user behavior, upon which they have some familiar tastes, will most likely have the same likes in the future. This humble yet potent concept has been augmenting the success of CF in practical contexts, such as Netflix, Amazon, and Spotify.

The majority of studies distinguish two principal types of CF algorithms: user-based and item-based. The user-based CF evaluates the preferences of sets of users with similar preferences and recommends only those items that users prefer within a particular group (Ricci et al., 2010; Schafer et al., 2007). The item-based CF, in turn, focuses on item similarity - this is where items similar to one another or frequently co-rated or co-interacted with many other items are suggested. An example would involve a scenario where a user had a similar interest to Inception, and another large proportion of users who enjoyed Inception also enjoyed Interstellar. In this case, Interstellar would be recommended to the user.

The CF methods have evolved over the years to become complex and model-based, such as Matrix Factorization (MF). They model users and items as a joint latent space, enabling the system to learn more complex patterns of preferences. Although effective, MF has one weakness, as the learning of good representations relies upon adequate interaction data.

To enhance the CF accuracy, particularly in sparse interaction settings, a growing body of research explores the use of hybrid models that incorporate additional content-based information, including item metadata, reviews, and user demographics. This trend has been further enhanced by the emergence of deep learning, which can capture non-linear and complex relationships and semantic signals in both item attributes and user interactions.

Such developments have further enhanced the performance of recommendation systems in terms of offering more accurate, varied, and significant recommendations, even in situations where they lack an extensive interaction history or are subject to semantic ambiguity.

With advancements in the field, deep hybrid models have become ubiquitous. They are used to reconceptualize the frontiers of CF, making it highly dynamic, scalable, and context-sensitive for practical applications.

* 1. **Matrix Factorization and Linked Data**

One of the most significant breakthroughs in overcoming the limitations of conventional memory-based Collaborative Filtering has been the adoption of Matrix Factorization (MF) methods, which provide a user-error interaction representation that is more general and scalable than a memory-based representation. These approaches are geared towards the discovery of hidden variables that cause observed ratings by learning low-dimensional embeddings of users and items, so as to decompose the user-item matrix. The MF methods have demonstrated good predictive capabilities and serve as the building blocks of most modern recommender systems.

Several extensions to the simple MF have been proposed in the hope of achieving better performance in sparse data scenarios. As an example, Natarajan et al. (2020) proposed RS-LOD (Recommendation System with Linked Open Data) and MF-LOD (Matrix Factorization with Linked Open Data) approaches, which incorporate external semantic information from an external knowledge base into user and item representations. The models enhance the quality of prediction in the cold-start and sparsity problem by including structured relational information.

On the same note, Nilashi et al. (2018) proposed a scalable matrix factorization algorithm that integrates Expectation-Maximization (EM) clustering with progressive Singular Value Decomposition (SVD). Such a hybrid model promotes the convergence rate and scalability of SVD, as it suits real-time recommendation tasks and yields high-accuracy results.

Da Silva et al. (2018) carried out a complete comparison of several MF methods, such as Non-negative Matrix Factorization (NMF), SVD, and Stacked Autoencoders, on the MovieLens 100k dataset with different sparsity in data. What their results highlighted was the trade-off between interpretability, accuracy, and robustness. More precisely, although the use of SVD overall had high performance, Autoencoders proved to be more adaptive when used in a sparsity scenario with greater impact.

All these contributions suggest that MF-based models can more accurately characterize the interactions in user-item pairs compared to the long-standing similarity-based techniques. In addition, the combination of Linked Open Data (LOD) and unsupervised learning methods (e.g., clustering and deep representation learning) has enabled exploring new opportunities in making matrix factorization more expressive and generalizable in real-world systems.

The problem is that, although matrix factorization is the most popular approach, it focuses solely on numerical ratings; thus, it may be unable to capture the complex semantic relations and context that may be embedded in user or item properties. This limitation has prompted the development of approaches in hybrid learning, as well as deep learning, which are discussed in the following sections.

* 1. **Review-Based and Hybrid Filtering Techniques**

As recommendation systems grew in size, it became apparent that simply using numerical data regarding user interaction, such as ratings, would not fully reflect the range of user preferences. To this end, the authors experimented more with the incorporation of auxiliary data that would complement the primary dataset of embedded features, such as textual reviews and user/item metadata. Such content-aware improvements are highly beneficial in reducing sparsity and enhancing the diversity and personalization of recommendations.

The purpose of review-based models is to leverage the sentiment and semantics conveyed in user-generated reviews. These models are enhancements of collaborative filtering, in that they incorporate the use of textual characteristics in the learning process. For example, Duan et al. (2022) presented a Review-Based Matrix Factorization method that enables learning both the rating and review data jointly, allowing for the discovery of latent preferences even when there is no explicit feedback. Similarly, a review-augmented matrix factorization model proposed by Isinkaye (2023) demonstrated increased accuracy in data sparsity settings by modeling the absence of ratings using textual context.

Another noteworthy work by Bobadilla et al. (2024) was a comparison of six matrix factorization models on four datasets, not based on prediction accuracy, but instead on novelty and diversity—the two aspects most important in user satisfaction in real-life systems. Their findings highlighted the importance of hybrid measures in determining model quality, beyond just using RMSE or MAE numbers.

To go beyond review-based enrichment, hybrid recommendation systems have become notable. These methods combine collaborative filtering and content-based filtering, often utilizing deep learning to integrate or merge various sources of information. Kiran et al. (2020) proposed a deep neural network (DNN) framework that integrates the user's demographics with the latent interaction data. Their results demonstrated a significant enhancement in dealing with cold-start problems, especially in the case of new users who lack historical interactions.

These hybrid models mark an important transition in the way recommender systems are designed: from the exclusive use of interactions towards a multi-modal learning model, in which side information (e.g., reviews, metadata, demographics) is considered a pertinent feature of item and user modeling. This advance has paved the way for the development of deep hybrid maps that incorporate encoders constructed by Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Transformer-based representations to learn more contextual representations.

* 1. **Deep Learning in Recommender Systems**

The question of fundamental transformation brought forward by deep learning has redefined the CF landscape, enabling the modeling of non-linear user-item interactions as well as more semantic representations. Li et al. (2024) and Martins et al. (2020) prove that models based on Multilayer Perceptrons (MLPs), Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and even Generative Adversarial Networks (GANs) can find complex patterns better than the traditional CF. Ibrahim et al. (2023) and Feng et al. (2020) noted that sensitive context-aware features can be helpful when combined with CF to increase relevance.

User reviews and other auxiliary signals, including those from social networks and knowledge graphs, have also been adopted to enhance representation learning. Wu et al. (2021) and Fallahi RahmatAbadi & Mohammadzadeh (2023) investigated them, demonstrating that deep or profound models trained in even more semantic contexts beat the classical CF models not only in predicting ratings but also in ranking items. Khoeini et al. (2020) further demonstrated how personalized deep representations can be applied in high-sparse environments to mitigate the problem of overfitting.

* 1. **Hybrid Deep Learning Models**

Hybrid CF models have progressed to advanced neural architectures. Fang et al. (2020) proposed CF-DNNF, a model combining the LSTM model and Deep Belief Networks, which yielded a significant reduction in RMSE and MAE. Latrech et al. (2024) proposed CoDFi-DL, which combines demographic filtering with deep learning, thereby improving cold-start resilience, particularly for new users. Shi et al. (2024) built upon this paradigm by introducing Graph Neural Networks (GNNs), which enable considerable richness in incorporating both user behavior and item-side information, and offer better ranking performance as well as robustness.

Specifically, the works of Pardo et al. (2024), who compared Autoencoders and Neural Collaborative Filtering (NCF), remain relevant when rating prediction is the task under consideration. Although they concluded that Autoencoders performed better according to RMSE, they noted that neither approach showed any awareness of semantic content, raising the requirement to hybridize again.

* 1. **CNN-based CF for Sparsity and Synonymy**

To directly address the issues of data sparsity and item synonymy, recent models have adopted CNN-based architectures that merge textual content and metadata. The work of Kim et al. (2016) suggested Convolutional Matrix Factorization, where probabilistic MF is combined with document-level features learned by CNNs, with a 3.92 percent increase over the MovieLens 1M. Liu et al. (2018) improved this idea by utilizing supervised CNNs that combine ratings and tag-level material to provide cooperative and content-based recommendations. Similarly, the approach proposed by Wu et al. (2019) presents a CNN-based projection mechanism that matches latent factors between reviews and ratings, thereby enhancing semantic coherence. Huo et al. (2017) utilized CNNs extended by Denoising Autoencoders, enabling CNNs to learn in extremely sparse conditions and be more resilient to noisy or incomplete data.

All the studies corroborate the idea that combining deep semantic representations with user-item embeddings enables not only the reduction of problems related to sparsity and synonymy but also the overcoming of issues that classical CF does not efficiently solve on its own.

This chapter has surveyed the advancements in recommendation techniques, which began with conventional CF models and evolved to deep learning-based hybrid or composite models. Although traditional approaches, such as those involving matrix factorization and autoencoders, have succeeded in informing how to model latent interactions, their evaluation results fall short in sparse contexts. In the meantime, more recent methods based on neural architectures and semantic extension, utilizing textual information, such as NCF and ConvMF, could provide a strong alternative.

There is, nevertheless, a definite research gap that is present. There are a limited number of models that effectively combine both collaborative and semantic learning into an unsupervised, harmonized construct. This disparity streamlines the request to the proposed ConCF model, which endeavours to narrow down these drawbacks by integrating the strengths of NCF and CNN in a scalable and content-wide network. What follows is the design of the methodology and model based on the observations described in the following chapter.

**Chapter 3: Methodology**

* 1. **Research Design and Framework**

This project utilizes an experimental model to optimize the performance of movie recommendation systems, with the ultimate aim of addressing two significant shortcomings of conventional Collaborative Filtering (CF) methods: data sparsity and synonymy. The paper methodically compared several concepts built on deep learning to determine the best approach for enhancing the accuracy of rating predictions and the quality of recommendations on MovieLens 1M.

The study utilizes an Autoencoder and Neural Collaborative Filtering (NCF) model to mitigate the problem of sparsity resulting from a high rate of missing user-item ratings. Such an attitude enables Artificial Neural Network (ANN)- based methods to learn latent user and item features with minimal amounts of data, thereby allowing them to recreate missing ratings more effectively compared to the classical matrix factorization algorithm.

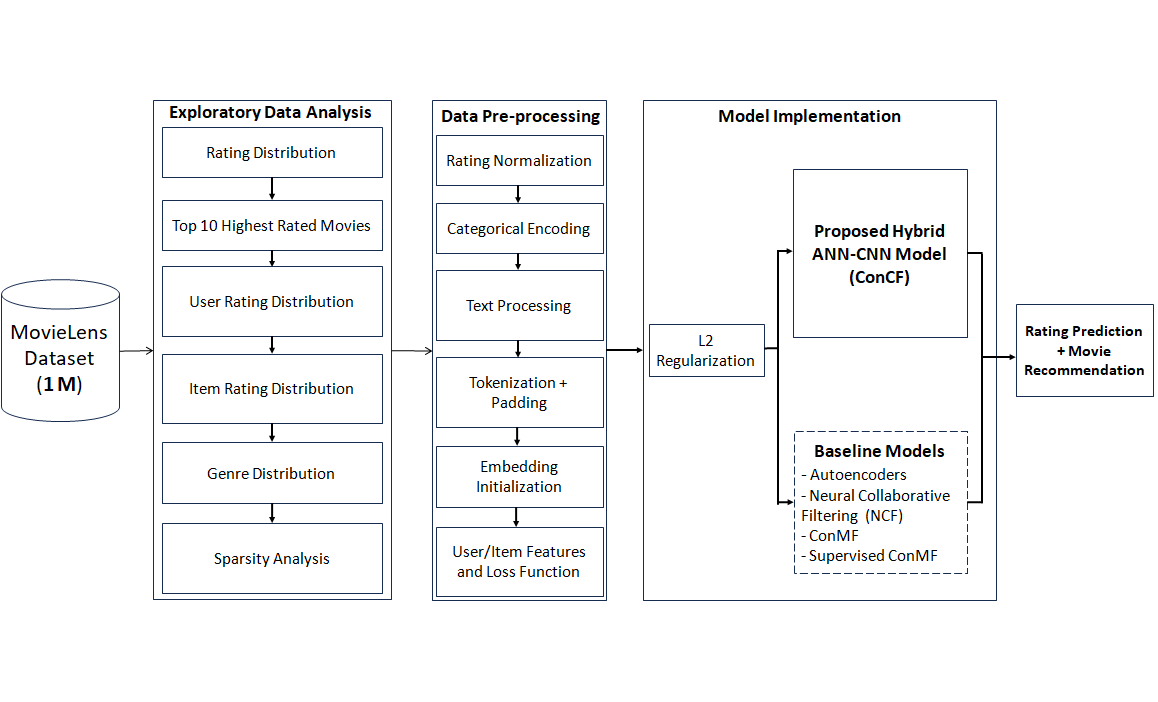
To deal with the synonymy issue, i.e., when semantically close objects (i.e., movies on a similar topic or plot) are considered dissimilar because there is limited data on the interaction between them, the study uses the models based on the Convolutional Neural Network (CNN), such as Convolutional Matrix Factorization (ConvMF) and Supervised ConvMF. These models utilize textual metadata (such as movie titles and genres) to represent items better and enhance the model's predictive capability regarding similar item recommendations.

Based on the above conclusions, it is possible to propose a new type of ANN-CNN hybrid model, named ConCF, that combines the advantages of the Autoencoder/NCF (regarding sparsity reduction) with the advantages of the CNN-based models (regarding the solution of synonymy issues). This hybrid construction integrates user and item interaction data, as well as the semantic type of content features, thereby improving the overall recommendation process.

### All models are trained and tested on the MovieLens 1M dataset without exceptions or violations, and performance is measured using both regression-type (RMSE) and ranking-type (Hit@1, Recall@5) metrics. The effectiveness of each model is also examined at different levels of sparsity to assess how it performs robustly in realistic recommendation conditions.

### **.**1 Methodology Flow Diagram****

The following diagram provides a general view of the end-to-end approach adopted in this study, which includes the preprocessed data, the model to be evaluated, and other relevant details:



**Figure 3.1**: **Schematic of a suggested methodology pipeline, which demonstrates the preparation of data, the design of the models (Autoencoder, NCF, ConvMF, Super-ConvMF, and ConCF), and their training on sparse inputs, and evaluation based on performance measures.**

### ****Dataset Description and Exploratory Data Analysis****

* + 1. **Dataset Description**

The MovieLens 1M dataset consists of:

1. 6,040 users have rated 1,000,209 movies.
2. Ultimate user profile information: age, gender, job, and zip.
3. Movie traits: titles and genres.
4. Tarife: according to a scale from 1 to 5.

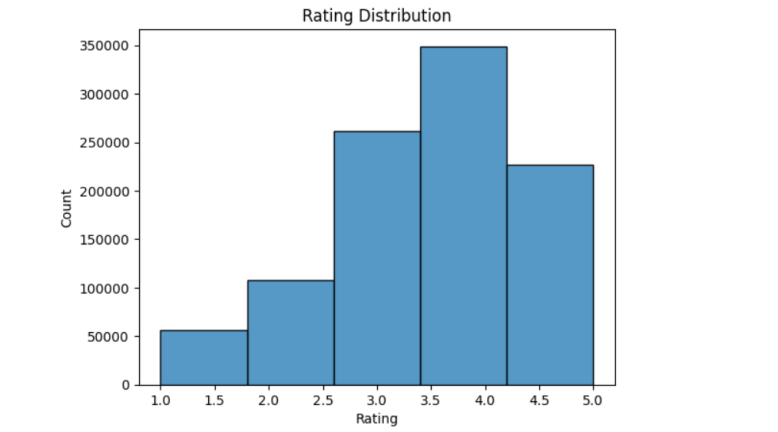
This dataset's wealth in both user-item interactions and side information provides the opportunity to apply collaborative, content-based, and hybrid recommendation systems.

* + 1. **Exploratory Data Analysis (EDA)**

A few exploratory calculations were made accordingly to gain more insights into the nature of the data to be used as the basis of the model design. The findings highlight the primary issues of rating altruism, item popularity disparity, and absolute sparsity, which directly impact the success of the recommendation model.

1. **Ratings Distribution:**

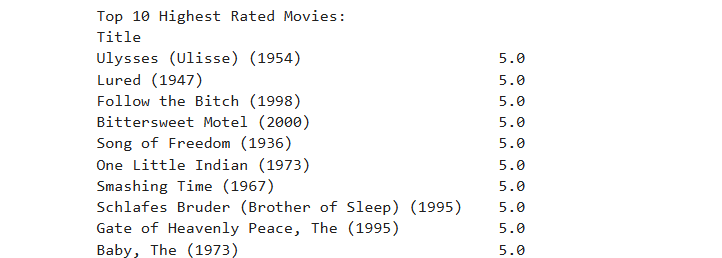
As illustrated in the histogram plot in Figure 3.2, most user rankings lie between 3 and 4, with a high concentration at 4. This implies a propensity to rate positively, indicating that there may be bias in the ratings, specifically in terms of rating inflation, which is a recognized bias in the case of user-generated datasets. This skewness may influence the learning process of models that consider all ratings to be equivalent, and it is essential to pay attention to normalization or bias-aware loss functions.



**Figure 3.2: Ratings Distribution**

1. **Top 10 Best-Rated Movies:**

In Figure 3.3, however, showing the top-rated movies, it is interesting that more than a dozen of them have each averaged 5.0. Some niche or cult movies predominate at the upper end of the ratings range. The rating, however, could only cover a minimal number of reviews, causing instability in variance and warranting that high total ratings of low-interaction channels should be treated with caution.



**Figure 3.3: The top-rated movies (all with an average rating of 5.0) include.**

1. **User Rating Distribution:**

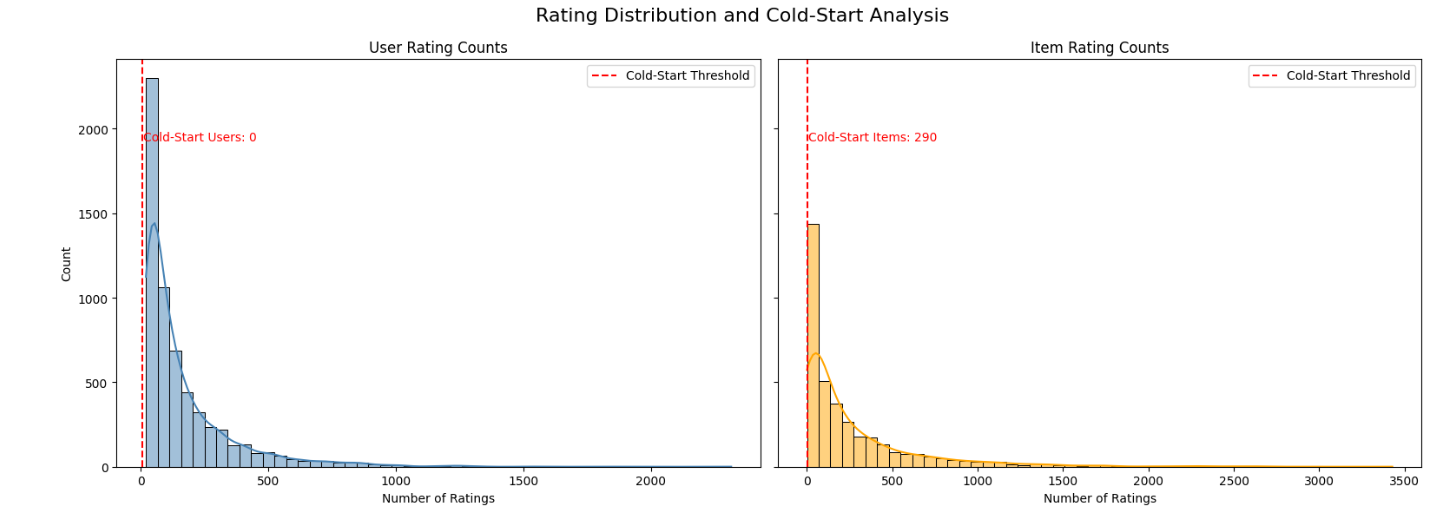
Figure 3.4 (left) reveals that the vast majority of users have rated a modest number of movies, and some users are very active. Important lessons learnt are:

1. All users have scored a minimum of 5 movies, which means that there are no hard cold-start users, and the collaborative filtering models can be very well implemented.
2. However, less engaged users can have noisy or undersampled preference vectors, and therefore, robustness methods such as dropout or user embedding regularization are often necessary.
3. One way to assess the distribution of the item rating is to divide it into four categories.
4. **Item Rating Distribution:**

As Figure 3.4 (right) depicts, a long-tail distribution exists, meaning that only a few popular items receive thousands of ratings, while most remain relatively untouched. This creates two problems:

1. Popularity bias: The Model can be over-recommending popular items without paying attention to low-rated content, which might also be relevant.
2. Cold-start items: There are about 290 movies with fewer than five ratings, and they cannot be as easily recommended by collaborative signals exclusively. This highlights the importance of adopting and integrating item metadata (e.g., genres, titles) for generalization in a semantic manner.

A combination of these two plots highlights an issue with data sparsity and an imbalance in popularity, which may impair performance within conventional recommendation systems.

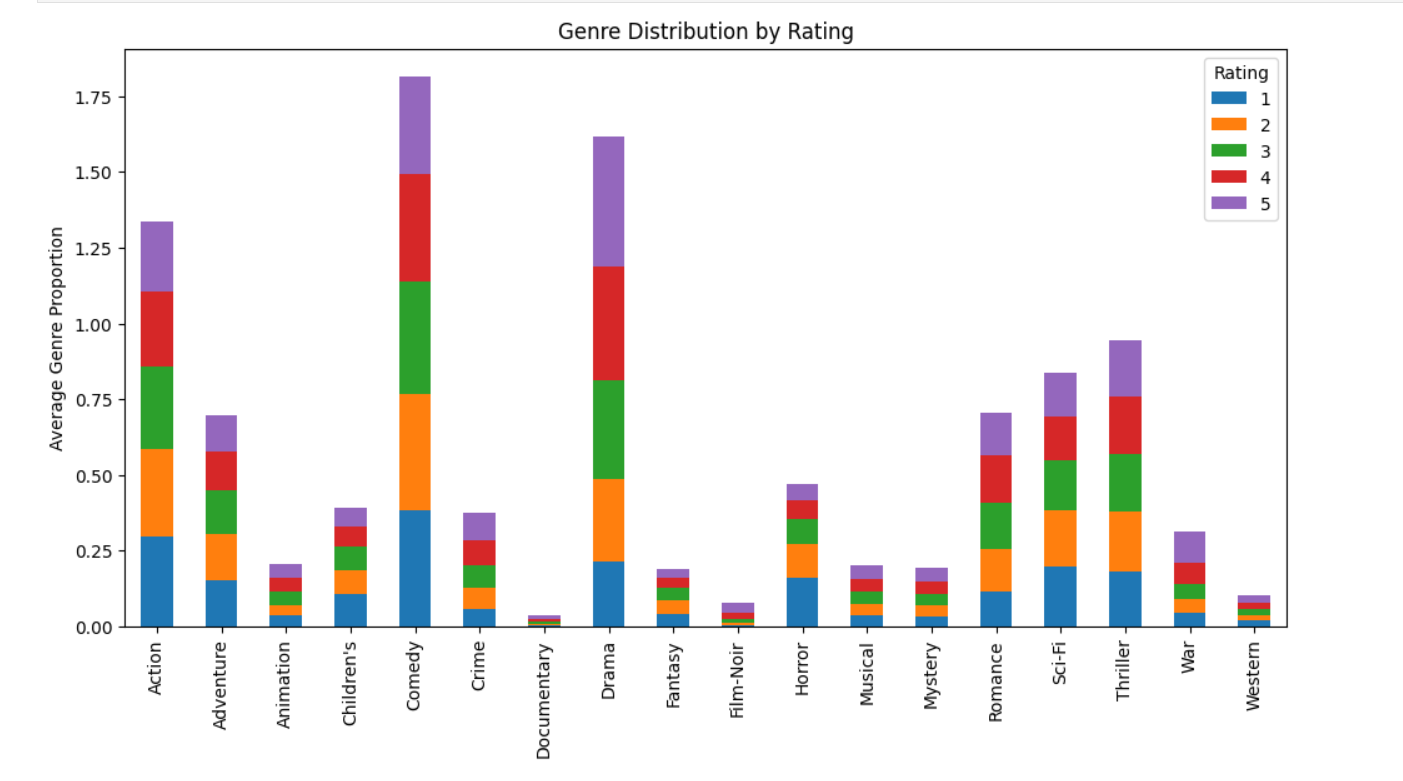


**Figure 3.4: User and Items Rating Distribution**

1. **Genre Distribution**:

As shown in **Figure 3.5**, genres like Drama, Comedy, and Action dominate the dataset. Notably, genres such as Documentary and Drama tend to receive higher ratings on average, indicating potential **genre-rating correlation**. This insight could inform genre-aware modeling strategies. Furthermore:

1. The distribution suggests an opportunity for **genre conditioning** in deep learning models.
2. Unbalanced genre presence may bias model training toward overrepresented categories.

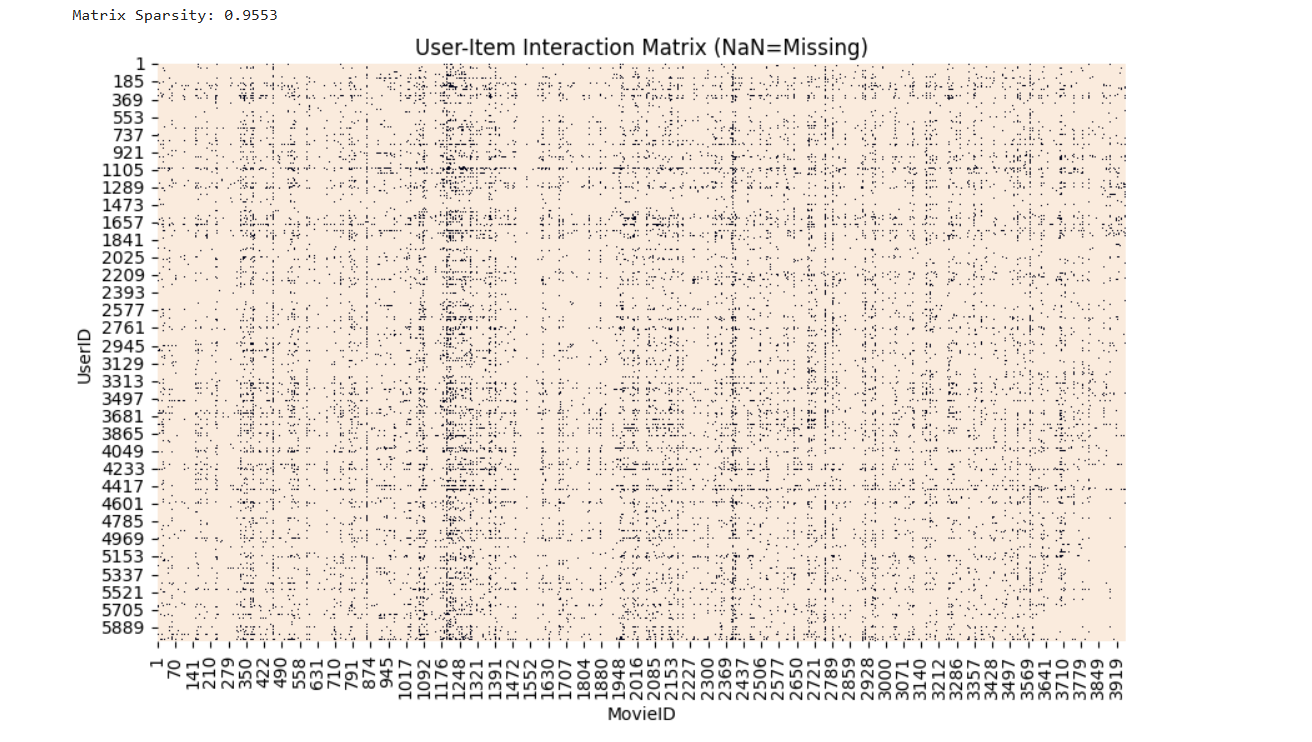


**Figure 3.5: Genre Distribution**

1. **Sparsity Analysis**:

The user-item rating matrix is visualized in Figure 3.6. With a sparsity rate of 95.53%, the dataset represents the standard problem of recommender systems in the real world: extreme data sparsity. Important notes are:

1. The presence of black dots dispersed represents the fact that few users and movies are associated with most of the interactions.
2. Cold-start and coverage problems are salient ways in which many users and objects do not have sufficient data with which to predict assertively.
3. This kind of extreme sparsity drives the demand for content-aware hybrid models that can learn with limited notices.

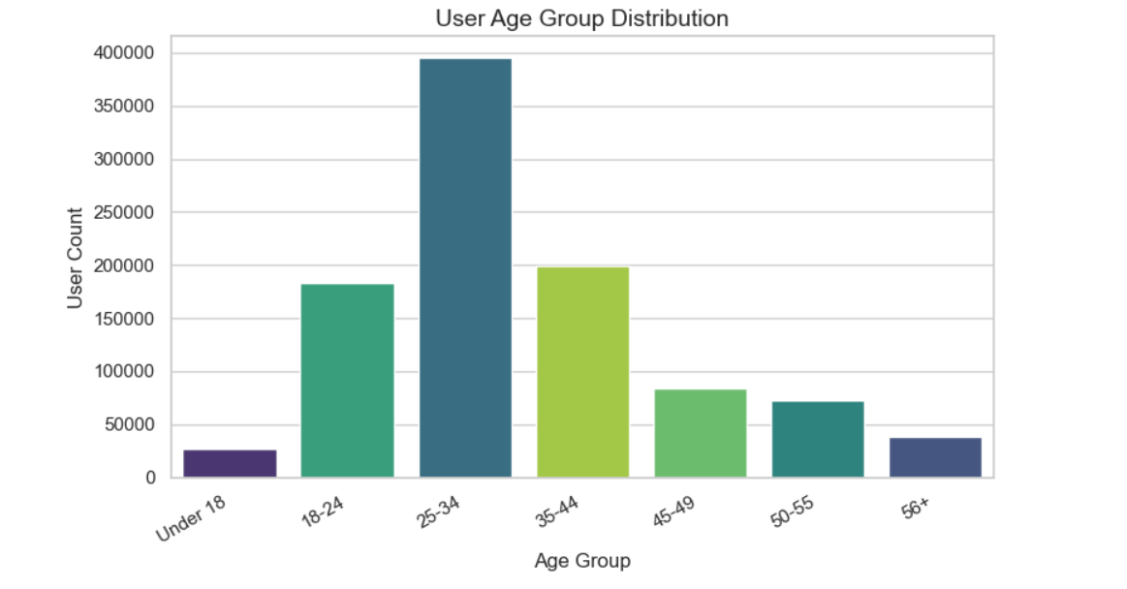


**Figure 3.6: Sparse User-Item ratings Matrix**

## Underlying these observations are the architectural choices made in the present thesis. To be more precise, the ConCF hybrid model was designed to address these practical issues by incorporating semantic metadata and latent interaction learning, making it particularly applicable to sparse and unbalanced datasets.

1. **User Demographic Analysis:**
2. **User Age Group Distribution:**

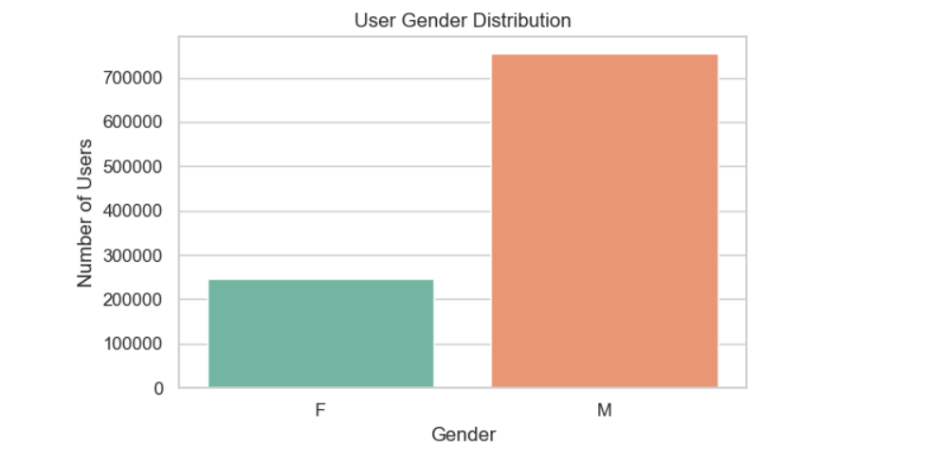
The age distribution plot has shown that most users are in the age category of 25- 34, followed by 35-44, and then 18- 24. This indicates that MovieLens is primarily populated by young and middle-aged adults who contribute the majority of ratings and are likely to exhibit specific attributes in the types of movies that attract more ratings and focus. The dynamics of age can also be used to make recommendations to address future cold-start situations and underrepresented groups of users who are older than 56 or younger than 18.



**Figure 3.7: User Age Group Distribution**

1. **User Gender Distribution:**

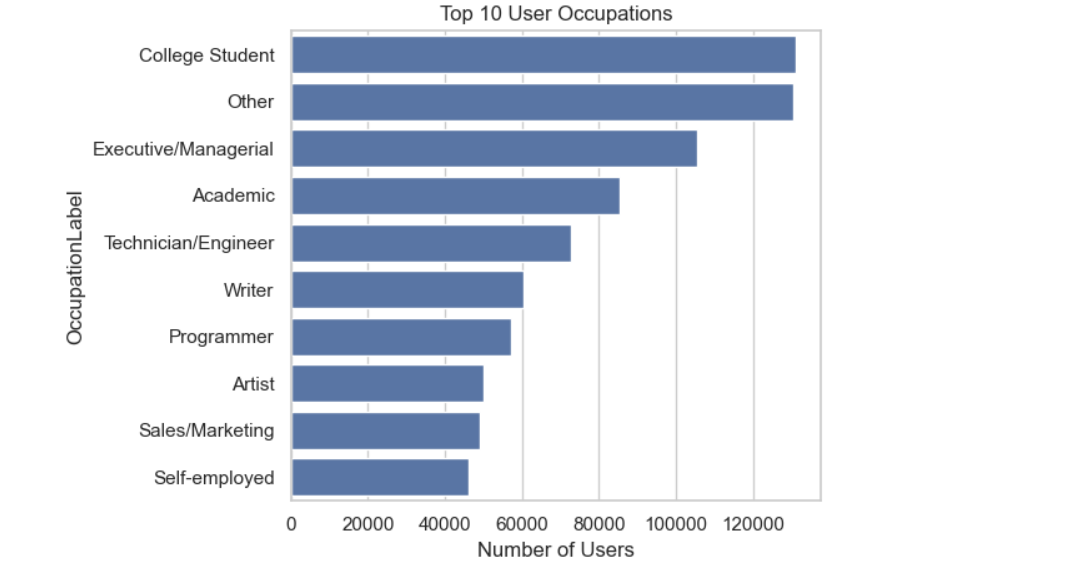
The sex breakdown is quite disproportionate, male users vastly outnumbering female users. This demographic bias indicates that recommendation models trained on this dataset inherit implicit gender biases, specifically, male-preferring genres and titles. Identifying this imbalance is needed for assessing model fairness and sensitivity to inclusion in future additions.



**Figure 3.8: User Age Group Distribution**

1. **Top 10 User Occupations:**

The top 10 user occupations plot indicates that most of the data in the set consists of college students or users classified as 'other'. The professionals, such as executives, engineers, and academics, are also well presented. This occupational understanding can be helpful in the development of contextual recommendation approaches and may also be useful in personalizing recommendations by occupation in any forthcoming improvements to user profiling.



**Figure 3.9: Top 10 User Occupations**

The analysis of demographic users reveals relevant trends by age, gender, and occupation, which are crucial for making further recommendations and developing the system. It can be noted that young adults (25-34) are dominant, the gender imbalance favors male users, and that college students are highly represented, showing possible bias and underrepresentation. These insights can play a crucial role in addressing cold-start issues and promoting fairness. Later implementations of the ConCF model may utilize demographic features as valid side data to produce more individualized, diverse, and sustained recommendation systems, particularly in settings with sparse or biased data.

## ****Data Preprocessing****

Preprocessing is a significant determinant of the success of recommender systems, particularly when handling sparsity and synonymy, which are the two major issues that this paper focuses on. In this section, we present preprocessing pipelines for various models, including Autoencoder, Neural Collaborative Filtering (NCF), ConvMF, Supervised ConvMF, and the proposed hybrid ConCF. The pipelines vary substantially, with a preference for using user-item interaction matrices, neural embeddings, and text-based metadata.

### ****Preprocessing for Autoencoder****

The autoencoder model requires the user-item matrix to be organized completely, with zeros indicating missing values. It solely aims to reconstruct the observed rating, in the course of learning compact user- and item-embeddings.

**Key Steps:**

1. ***Label Encoding:*** *Categorical features, such as genres and movie titles, were encoded into integers using LabelEncoder.*
2. ***User-item rating Matrix:*** *A pivot table was developed, where the rows represent users, the columns represent movies, and the values represent ratings. There were zeros where ratings were missing.*
3. ***Rating Normalization:*** *MinMaxScaler was applied to normalize ratings to a [0, 1] scale, facilitating stable training of the data by neural networks.*
4. ***Masked Loss Function:*** *The customized loss function (masked\_mse) was developed to add only the observed rating to the training loss. This does not allow the model to penalize unobserved entries that are zero valued.*



This way, the model learns informative latent space while dealing with sparsity by reconstruct missing data.

### ****Preprocessing for NCF, ConvMF, Supervised ConvMF, and ConCF****

### *Step 1: Dataset Cleaning & Merging*

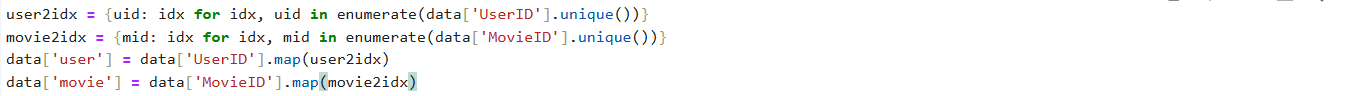
1. *I dropped rows with null values from the ratings, users, and movies datasets.*
2. *Merged of datasets on UserID and MovieID.*
3. *Ratings out of the valid [1, 5] range have been removed.*

***Step 2: Text pre-processing for Item metadata***

1. *Genres are cleansed by stripping of delimiters (|) and non-alphabetic characters, and lowercasing.*
2. *As a turn, titles as well were scrubbed.*
3. *The two are integrated into a single text string to be input for a CNN-based model.*
4. *This enables the model to be trained on semantically rich metadata, allowing it to identify latent themes that can overcome the problem of synonymy.*

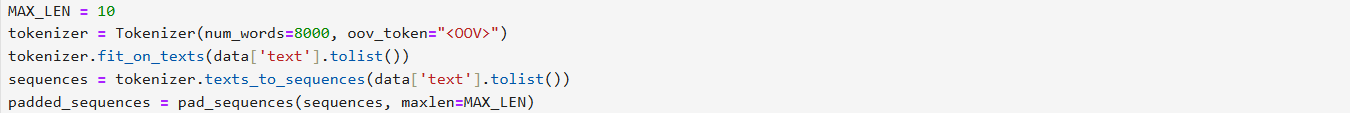
***Step 3: ID Mapping for Embedding Layers***

1. *User and movie IDs have been mapped to separate embedded indices for use as inputs to the embedding layer.*



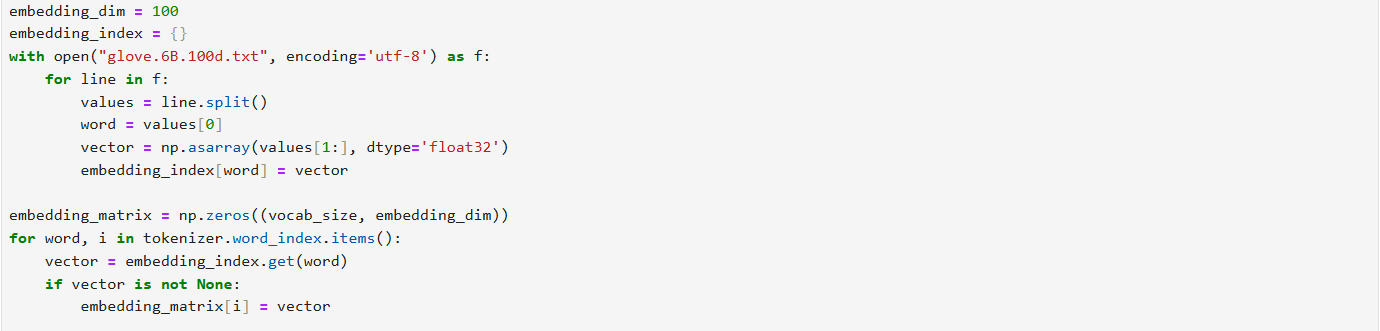
#### ****Step 4: Tokenization and Padding****

Each text was tokenized into words (including titles and genres) with the help of the Tokenizer and padded to a fixed sequence length to ensure identical input sizes for CNNs.



#### ****Step 5: Pretrained Word Embedding (GloVe 100d)****

*To utilize more semantic information from the text of items, pretrained GloVe embeddings (100 dimensions) were loaded into memory and used to initialize the CNN embedding layer. The words, which were not included in the GloVe vocabulary, were given a zero vector.*



#### With pretrained embeddings, there is already a lead on achieving semantic similarity, a plus to the performance, particularly with the sparse or cold-start items.

#### ****Step 6: Final Feature Preparation****

*The final feature set for model training included:*

### *X user: Coded user indices*

### *X m movie: Coded movie indices*

### *X\_text: Pad and tokenised text sequences*

### *y: ratings on the [0, 1] scale (i.e. ratings/ 5.0)*

### ****Comparison with Supervised ConvMF Preprocessing****

Although this paper is based on in-dataset metadata (titles and Genres), the Supervised ConvMF model employs external databases, such as IMDB, to obtain complete descriptions and tags. The important differences are:

**Table 3.1: Comparison with Supervised ConvMF Preprocessing**

| **Aspect** | **This Study (ConCF)** | **Supervised ConvMF (Liu et al., 2018)** |
| --- | --- | --- |
| Text Source | Title + Genre (from MovieLens) | Description documents (e.g., IMDB) |
| Tag Supervision | Not used | Tags used for multi-label classification |
| Embedding Initialization | Pretrained GloVe 100d | Random or learned from scratch |
| Preprocessing Scope | In-dataset only | External data crawling |
| Tokenization Strategy | Keras Tokenizer + padding | Bag-of-words or CNN word matrix |
| Loss Function (Text) | Rating regression | Joint rating + tag prediction |

### ****Summary of Preprocessing Pipelines****

**Table 3.2: Summary of Preprocessing Pipelines**

| **Step** | **Autoencoder** | **NCF / ConvMF / ConCF** |
| --- | --- | --- |
| Rating Normalization | √ (MinMaxScaler) | √(Rating / 5.0) |
| Categorical Encoding | LabelEncoder | Embedding indices |
| Text Processing | **×** | Title + Genre → Cleaned text |
| Tokenization + Padding | **×** | Keras Tokenizer + MaxLen = 10 |
| Embedding Initialization | **×** | Pretrained GloVe (100d) |
| User/Item Features | User-Item Matrix | User/Movie ID Mapped |
| Loss Function | Masked MSE | MSE (or hybrid if multi-task) |

In the study, the preprocessing procedures adopted are well-adjusted to the requirements of various models. With sophisticated approaches to NLP that utilize embeddings on CNN-based networks, normalization, and masking in matrix-based Autoencoders, the pipelines will generate durable learning from sparse data. These steps also place the proposed hybrid ConCF model at an advantage over traditional methodologies, such as Supervised ConvMF, particularly when dealing with the same data shortages.

## ****Baseline Models****

### To benchmark the performance of the proposed ConCF model, the study implemented four latent models from a widely recognized literature on recommendation systems. These models are chosen because of their success in addressing the fundamental issues of collaborative filtering, namely data sparsity and item synonymy. Each of the models has been trained under identical preprocessing conditions and tested over various sparsity levels, with RMSE used as the measure of accuracy.

### These are the baseline models that handle sparsity, synonymy, or a combination thereof using collaborative and content-based methods. Therefore, it is also appropriate to compare them directly with the proposed hybrid concept of ConCF. Content-aware (ConvMF and Supervised ConvMF) and collaborative filtering (Autoencoder and Neural Collaborative Filtering (NCF)) based models learn latent representations of user-item as a method to alleviate sparsity, whereas collaborative filtering based models (Autoencoder and Neural Collaborative Filtering (NCF)) learn latent user-item interactions to mitigate data sparsity, whereas content-aware based models (Convolutional Matrix Factorization (ConvMF) and Supervised ConvMF) use semantic features of the movie metadata to capture item similarities

### ****Autoencoder (AE)****

1. **Theoretical Overview:**

Autoencoders are non-supervised artificial networks that train themselves to reproduce their input. In collaborative filtering, they estimate the user-item interaction matrix and make missing entries by learning a latent representation.

1. **Summary of implementation:**

**We built a symmetric deep autoencoder consisting of three layers in the encoder and decoder without using dropout to regularise. The input parameters are the normalized version of the user-item ratings, and the loss function is the masked average squared error**

1. **Training Strategy:**

**Learned on dense and sparse (20-80%) rating matrix using early stopping and Adam optimizer (lr= 0.001).**

* + 1. **Neural Collaborative Filtering (NCF)**

1. **Theoretical Overview:**

NCF applies a combination of matrix factorization and deep neural networks to learn the complex nonlinear interactions between users and items.

1. **Summary of Implementation:**

The concatenation of the user and movie embeddings is then fed as input into two types of dense layers, where the activation functions are ReLU and dropout. The senior forecast is a score that is sigmoid scaled, meaning that rates are converted to a normal distribution.

1. **Training Strategy:**

Ratings were normalized to the range [0,1]. The model used embedding dimensions of ϕᵤ = 100 for users and ϕᵥ = 10 for movies. Training was performed using the Mean Squared Error (MSE) loss function and optimized with the Adam optimizer at a learning rate of 0.001.

### ****Convolutional Matrix Factorization (ConvMF)****

1. **Theoretical Overview:**

ConvMF improves PMF by replacing the item latent vector with a CNN-based embedding composed of item metadata (e.g., title and genre text). This enables the model to represent synonymy at the item level.

1. **Summary of Implementation:**

The model incorporates user embeddings on tokenized text; user embeddings are learned, and item embeddings are generated using CNN. The CNN output is consequently mapped to a common latent space and is combined with the user vector through a dot product.

1. **Training Strategy:**

We used pretrained GloVe (100d) embeddings, used a Conv1D model trained with MSE loss and activated with ReLU.

### ****Supervised ConvMF (Convolutional Matrix Factorization)****

1. **Theoretical Overview:**

Supervised ConvMF adds multi-task learning to ConvMF by jointly predicting ratings and genres. This helps to normalize the CNN path and increases the semantic representation quality.

1. **Summary of Implementation:**

Besides the ConvMF architecture, there is a second output layer that predicts genre tags (a multi-label classification). This model features a CNN branch common to both rating prediction and tag classification.

1. **Training Strategy:**

Trained together in the MSE rating and binary cross-entropy tag prediction. GloVe embeddings were recycled and not changed by training. The values of the loss weights were assigned a value of 1.0 to both outputs.

**Table 3.3 – Baseline Model Comparison Overview (as reported in Ref.** Liu et al., (2018, p. 1850018-13))

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Model** | **User Embedding** | **Item Embedding** | **Text Input** | **Extra Output** | **Loss Function(s)** | **Tags Used** | **Word Embedding** |
| Autoencoder | – | – | No | No | Masked MSE | No | No |
| NCF | 100 | 10 | No | No | MSE | No | No |
| ConvMF | 100 → 10 | CNN → Dense(10) | Title + Genre | No | MSE | No | GloVe 100d |
| Super-ConvMF | 100 → 10 | CNN + Dense(100, 10) | Title + Genre | Tags (multi-label) | MSE + Binary Cross-Entropy | Yes | GloVe 100d |

The Super-ConvMF model has the advantage of performing multi-task learning and aligning user-item ratings with textual semantics and genre supervision. This enhances sparse generalization and gives interpretability of tag predictions. In your re-implementation, fairness in the evaluation of comparative analysis is provided by having a single preprocessing pipeline and internal metadata applied to all models.

* 1. **Proposed Hybrid Model: ConCF (Collaborative Neural Content Fusion)**

In this section, the proposed framework, ConCF (Collaborative Neural Content Fusion), is presented as a hybrid architecture to address two challenges that have long existed in collaborative filtering: sparsity and synonymy. The model combines the power of Neural Collaborative Filtering (NCF) in modeling latent user-item interactions with Convolutional Neural Networks (CNNs) in extracting semantic relations from item metadata (e.g., titles and genres). NCF generates user-item interactions through learned embeddings but cannot take text as input. The CNNs, in turn, are effective at localizing the semantics of unstructured data. Integrating both, ConCF utilizes structured behavioral data and rich semantic knowledge in parallel, thus enabling improved predictive accuracy as well as generalization at differing extents of data sparsity and item synonymy.

### ****Motivation and Design Rationale****

Most users accessed only a small fraction of items, which makes it challenging to learn the preferences of users in collaborative filtering. Additionally, traditional models fail to identify the semantic equivalence of items (e.g., similar movies with different titles), resulting in gaps in synonymy.

1. In order to temper these concerns, the ConCF model incorporates:
2. Neural collaborative filtering branch to simulate the latent space interactions.
3. A CNN-based semantic encoder - to obtain textual features of the movie metadata (titles and genres).
4. A fusion layer that integrates collaborative and semantic signals to make robust rating predictions.

This hybrid approach enables generalizing the model in situations with limited user and item interactions, suggesting items with stronger semantically based relevance.

* + 1. **Model Architecture**

The offered ConCF model includes three modules with interconnections:

1. **User and Movie Embedding Layer:**
2. Each user u is represented by an embedding vector eᵤ ∈ ℝ¹⁰⁰.
3. Each movie m is assigned an embedding vector eₘ ∈ ℝ¹⁰.
4. The learnings of such embeddings are done through training using observed user-item interactions.
5. **Textual Feature Extraction via CNN:**
6. The concatenated movie title and genre are tokenized and passed through an embedding matrix initialized with 100-dimensional GloVe vectors.
7. The combined title and genre of the movie is tokenized and run on an embedding matrix of dimension 100 initialized on GloVe vectors tₘ ∈ ℝᵈ, capturing high-level textual features.
8. **Fusion and Prediction Layer:**
9. The three vectors eᵤ, eₘ, and tₘ are concatenated into a unified feature vector:
10. z = [eᵤ ∥ eₘ ∥ tₘ]
11. This fused representation is passed through one or more fully connected layers with nonlinear activations (e.g., ReLU).
12. The final output r̂ᵤ,ₘ is produced using a sigmoid activation function to ensure the predicted rating falls within the normalized range [0, 1]:
13. r̂ᵤ,ₘ = σ(f(z))
    * 1. **Mathematical Formulation and Prediction Calculation**

The proposed ConCF model computes the predicted rating r̂ᵤ,ₘ for a user u and movie m by combining three sources of information:

* 1. User Embedding eᵤ ∈ ℝ^{dᵤ}
  2. Movie Embedding eₘ ∈ ℝ^{dₘ}
  3. Text-based Semantic Vector tₘ ∈ ℝ^{dₜ} extracted via CNN over item metadata

These are concatenated to form the input vector:

z = [eᵤ ∥ eₘ ∥ tₘ] ∈ ℝ^d, where d = dᵤ + dₘ + dₜ

Forward Pass through the MLP

1. **Hidden Layer:**

h₁ = ReLU(W₁ ⋅ z + b₁), where W₁ ∈ ℝ^{h×d}, b₁ ∈ ℝ^h

1. Output Layer (before activation):

s = W₂ ⋅ h₁ + b₂, where W₂ ∈ ℝ^{1×h}, b₂ ∈ ℝ

1. **Apply Sigmoid Activation:**

r̂ᵤ,ₘ = σ(s) = 1 / (1 + e^(−s)) ∈ [0, 1]

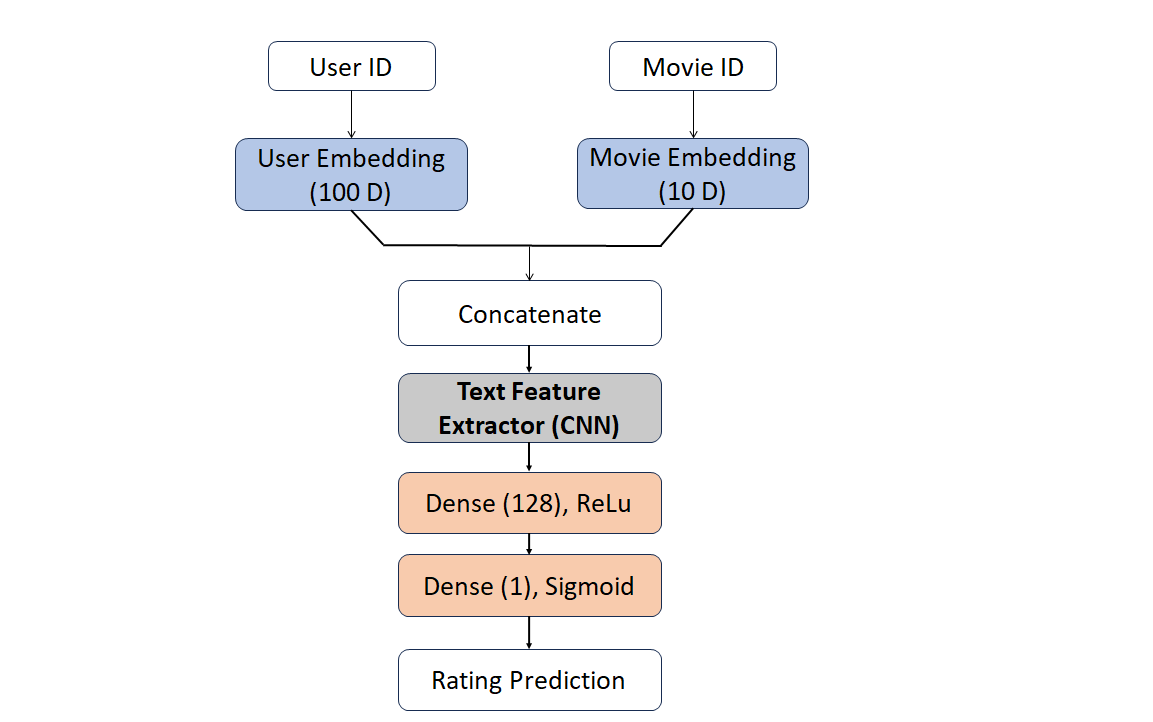
1. **Denormalize (if ratings scaled to [0,1]):**

r̂ᵤ,ₘ\_final = r̂ᵤ,ₘ × Rₘₐₓ, where Rₘₐₓ = 5 (if ratings range from 1 to 5)

**Table 3.4: Symbol Description**

|  |  |
| --- | --- |
| Symbol | Description |
| eᵤ | User embedding vector (e.g., 100-dim) |
| eₘ | Movie embedding vector (e.g., 10-dim) |
| tₘ | CNN-derived item text vector (e.g., 50-dim) |
| z | Concatenated feature input |
| W₁, W₂ | Dense layer weights |
| b₁, b₂ | Dense layer biases |
| σ(·) | Sigmoid activation function |

### ****Model Diagram****

**

**Figure 3.10: Architecture of the proposed ConCF model combining user/movie embeddings and CNN-based semantic content representation.**

### ****Implementation Summary****

To implement a suggested ConCF (Collaborative Neural Content Fusion) framework, a compound deep learning structure was developed using TensorFlow and Keras, which closely matched the theoretical one. Implementation was implemented in modules corresponding to each functional block of the architecture:

1. **User and Movie Embeddings**:

Users and Movies have been encoded as low-dimensional embedding vectors with dimensions of 100 and 10, respectively. These embeddings learn latent interaction signals from the historical user-item ratings.

1. **Textual Semantic Module (CNN):**

Item-level metadata consisted of movie titles, genre, etc., and was preprocessed in the form of padded sequences of tokens and encoded with pretrained GloVe vectors. It utilized a one-dimensional convolutional layer with multiple filters, and subsequently, global max-pooling generated a semantic vector of fixed length to describe each item.

1. **Fusion Layer:**

The concatenated fusion of user embedding, movie embedding, and semantic vector, as derived using CNN, was made as a combined feature vector. The same vector was processed by several dense layers with ReLU activation, batch normalization, and dropout regularization, allowing for the nonlinear combination of collaborative and semantic signals.

1. **Prediction Output:**

The final rating prediction was generated using a sigmoid-activated dense layer, producing normalized ratings within the range [0, 1]. These predictions were denormalized as needed to align with the original rating scale (e.g., [1, 5]) during evaluation.

1. **Optimization and Training:**

The The model was trained using the Adam optimizer with a learning rate of 1×10−31 \times 10^{-3}1×10−3, and the loss function used was Mean Squared Error (MSE). Key regularization strategies included dropout, early stopping, and learning rate scheduling. The training process was conducted across four levels of sparsity (20%, 40%, 60%, and 80%) to evaluate the model's robustness with limited interaction data.

The implementation was done in a way that was faithful to the architectural design, while also allowing for flexibility in experimentation. RMSE and Top-N recommendation metrics confirmed that the hybrid architecture achieved a higher performance that the baseline models.

# **Hyperparameter Configuration and Tuning Strategy**

To ensure optimal model performance and prevent overfitting, a systematic combination of empirical heuristics and light tuning was used to select the model’s key hyperparameters. While extensive grid or random search was not required due to prior research knowledge and computational constraints, each component of the hybrid ConCF model was carefully calibrated based on existing literature, ablation experiments, and domain-specific insights.

## Model Architecture Hyperparameters:

**Table 3.5** **Model Architecture Hyperparameters**

|  |  |  |  |
| --- | --- | --- | --- |
| **Component** | **Hyperparameter** | **Value** | **Rationale** |
| Embedding Layer | User embedding size | 100 | Matches latent factor dimension (ϕᵤ=100); enables user diversity |
| Embedding Layer | Movie embedding size | 10 | Low-rank (ϕᵥ=10) sufficient for item representation due to CNN fusion |
| CNN Branch | Embedding dim (text) | 100 | GloVe 100d pre-trained vectors used to encode semantics |
| CNN Branch | Conv1D filters | 256 | Empirically found to balance expressiveness and overfitting |
| CNN Branch | Kernel size | 5 | Effective in extracting short n-gram features in tag/title descriptions |
| CNN Branch | Kernel regularizer | L2 (1e−4) | Prevents overfitting during convolution |
| MLP Layers | Dense units | 128 | Standard in NCF; captures high-order interactions |
| MLP Layers | Dropout rate | 0.4 | Reduces overfitting; tested between 0.2–0.5 |
| Output Layer | Activation | Sigmoid | Scales prediction to [0,1] range (normalized ratings) |
| Optimizer | Adam | lr=1e−3 | Adaptive learning rate; baseline optimizer for deep models |

## Training Configuration and Dynamic Learning:

**Table 3.6** **Training Configuration and Dynamic Learning**

|  |  |  |
| --- | --- | --- |
| **Component** | **Setting** | **Justification** |
| Epochs | 50 | Empirically found to converge; early stopping avoids overfitting |
| Batch Size | 512 | Balances memory usage and convergence stability |
| EarlyStopping | Patience=5 | Prevents unnecessary training once validation loss plateaus |
| LearningRateScheduler | Decay after 20 epochs | lr = lr × 0.8 after 20 epochs to fine-tune weights |
| Validation Split | From train-test partition | 5 cross-validation runs to average generalization performance |

## Tuning Logic Summary:

Instead of an exhaustive search of hyperparameters (e.g., Bayesian Optimization or Keras Tuner), a more focused approach was used:

1. **Domain-Informed Initialization:** Embeddings and CNN filters dimensions have been initialized based on some previous successful implementation.
2. **Empirical Calibration:** Dropout and learning rate, among other parameters, were tried in small bounded ranges in small training cycles.
3. **Scheduling of Learning Rate:** It was also added by using a decaying schedule (lr = lr x 0.8 at epoch 20) to converge.
4. **Future Scope for Full Tuning:**

To enable long studies or production use within an industry, this structure is compatible with automated tuning libraries, like Keras Tuner or Optuna. It would enable a grid/random/Bayesian search of the dropout rates, sizes of the embeddings, the width of the kernels, and the dimensions of the MLPs, thereby customizing the ConCF model to various databases or application domain constraints.

## Evaluation Setup

To evaluate the performance of the ConCF model, training was conducted at various sparsity levels (20%, 40%, 60%, and 80%) using the same data preprocessing methods as the baseline models. The model was tested using the same splits as the Autoencoder, NCF, ConvMF, and Super-ConvMF to ensure a consistent and fair comparison. This section includes brief descriptions of the evaluation metrics used:

1. **Root Mean Squared Error (RMSE):** A widely used regression metric that measures the average magnitude of error between predicted and actual ratings. It is calculated as:

RMSE = √(1/n \* Σ(yᵢ - ŷᵢ)²), where yᵢ is the actual rating, ŷᵢ is the predicted rating, and n is the number of predictions. A lower RMSE indicates higher predictive accuracy.

1. **Recall@5:** A ranking metric that evaluates the proportion of relevant items retrieved in the top 5 recommendations. It is calculated as:

Recall@5 = (Number of relevant items in Top-5) / (Total number of relevant items).

**1-best Accuracy (Hit@1):** This metric measures whether the top-ranked item matches the user's actual preferred item. It is defined as:

Hit@1 = 1 if the top predicted item is correct, otherwise 0. The overall score is the average over all users.

* + - 1. **Justification for 20% Sparsity Benchmark**

Although the ConCF model was tested at a wide range of sparsity levels (20%, 40%, 60%, 80%), 20 percent sparsity (i.e., 80 percent of the training data) was chosen as the primary performance measure. At this level, the amount of training data is the highest, enabling every model to demonstrate its capabilities to the fullest and showcase its real potential in a favorable environment.

Methodologically, the critical point of choosing the comparative benchmark of 20% sparsity is that the predictive power of the model is not overrepresented by data sparsity, which may hide architectural effects and overemphasize random noise. In such an environment, differences in performance indicate more about the model design's abilities than noise caused by inadequate data.

Additionally, such a decision is consistent with the procedure followed by previous studies [e.g., Kim et al. (2016) and Liu et al. (2018)], where simulators also were tested in the low-sparsity regime to ensure the measurement of their basic learning capacity, after which they were put under the stress test at higher levels. In the thesis, the ConCF model had the lowest RMSE of 20 in terms of sparsity, which justified it as the most prominent basis of comparison regarding overall performance with the baseline models.

While common in practical applications, greater sparsity levels may be driven by platforms trying to lower the sparsity by exploiting implicit signals (i.e., clicks, views, dwell time), but a best-case approximation to such richer settings is the 80 percent known data benchmark. That is why 20% is not only realistic but also methodological, in the sense that evaluating the optimal, best possible performance of a model is appropriate.

### ****Model Limitations and Design Trade-offs****

Although the proposed ConCF model exhibits excellent performance in empirical studies, it has several limitations that warrant discussion to further improve and implement it in real-life systems.

The first downside of the model is that it relies heavily on pre-trained word embeddings (GloVe) to derive the semantic relationships in movie metadata. This method is computationally efficient, but it assumes the static meaning of words, without considering contextual variants that may exist in varied movie descriptions. Consequently, some low-level semantic relations between items may go uncaptured.

Second, the combination of a hybrid model of ConCF makes its architecture more complex, which can restrict its interpretability and scalability to large-scale implementations. Even though the hybrid scheme of Neural Collaborative Filtering (NCF) and Convolutional Neural Networks (CNNs) improves recommendation accuracy, this approach can result in large parameter sets if the training data is scarce or highly sparse, thus leading to the overfitting effect.

Moreover, only the MovieLens 1M dataset was used for training and testing the model. Despite this dataset being broadly applicable and representative in terms of genre and user activity, the extent to which ConCF applies to other spheres (e.g., e-commerce, music) has not yet been tested. Transferability can be restricted in cases where the quality of metadata describing different items, user behavior, or their sparsity patterns differs.

The following limitation is the use of ratings as ground truths on relevance. Ratings are subject to varying degrees of social bias, interface design, or rating inflation in practice, so they may not necessarily be a valid index of user satisfaction. The more subtle forms of feedback (such as dwell time, skips, and replays) have the potential to enhance the reality of training goals.

* + - 1. **Design Trade-offs: Word Embeddings and Model Architecture**

The decision to use GloVe embeddings rather than contextualized embeddings, such as BERT, when designing ConCF is at least partially an intentional compromise between semantic richness and computational memory requirements. Although BERT provides better contextual knowledge and dynamic embeddings, it is computationally intensive. It is more commonly used as a fine-tuning-based solution, implying that no real-time application or resource-scarce setting can rely on BERT. GloVe, in turn, provides static embeddings that effectively reflect global word co-occurrence statistics, thereby enabling the CNN module to reveal informative semantic patterns of manageable complexity. On the same note, architecture does not rely on extremely deep or attention-specific models to achieve scalability and interpretability. This is a design decision focused on achieving high performance without complicating deployment and training processes.

In brief, although ConCF can counter severe weaknesses such as sparsity and synonymy using hybridization, new constraints are generated that concern the complexity of architecture, the meaning of context, and dependence on the dataset. Future research should simplify the model, investigate more informative feedback on nutrition, and test extrapolation in other areas and languages.

* + 1. **Transferability of Results**

The transferability of experimental results. The appropriateness of a model's performance when measured with one dataset or scenario for use on other datasets, domains, or user populations. The ConCF model in this study was trained and tested on the MovieLens 1M dataset—the most famous benchmark in movie recommendation research, which contains structured rating data, item metadata (i.e., genres, titles), and user features (i.e., age, gender, occupation). Although this dataset serves as a comprehensive background for model development and comparison, several factors affect the generalizability of the findings.

1. **Qualities of the Dataset:**

MovieLens 1M is a well-balanced and structured dataset of ratings from active users. However, recommendation systems may need to deal with noisier, less cleaned-up, and more dynamic data. Therefore, the ConCF model, albeit demonstrating excellent results in numerous sparsity settings (20 80%), might not be successfully transferred to such areas as e-commerce, music, or news recommendation; this could involve adjustments in the content encoding (e.g., the replacement of movie genres with product descriptions or tags).

1. **Model Modularity:**

A key strength of the ConCF architecture is its modularity. It is possible to retrain the CNN component with fresh domain-specific text (e.g., item descriptions, user reviews), and it is also possible to re-initialize the embedding layers for any user-item combination. This makes the model one that is amenable to cross-domain deployment, given due preprocessing and retraining for each domain.

1. **Pretrained Embeddings and Language Dependency:**

From GloVe embeddings, specific dependence on the English language and the quality of textual representation of items emerge. The CNN component may exhibit different performance in cases where there is limited textual metadata, multilingual content, or informal text created by the user. Replacing GloVe with domain-trained or contextual embeddings, such as BERT, would be an option to enhance flexibility, although further experimentation would still be needed in the future.

1. **Evaluation Protocols and Metrics:**

#### It was analyzed in terms of RMSE, MAE, Recall@5, and Hit@1, which are the most commonly used metrics in the scholarly community. Commercial systems are, however, commonly optimised for long-term engagement, diversity, or profit, all of which may need retraining on different objective functions or rank-based evaluation procedures.

#### Practical Constraints:

### Latency and Scalability considerations also influence Real-world transferability. Though the ConCF model allows a trade-off between performance and complexity, to use it in a real-time recommendation engine, it would be necessary to optimize by compressing the model or using inference pipelines.

### Model Comparison Overview

This section provides an overview of the most important design aspects and the focus of the problem for all the baseline and proposed models employed in this research, in order to make the comparative analysis more structured and fair. We trained all models with an identical preprocessing pipeline, feature embedding dimensions, and evaluation criterion (RMSE). In the comparison, the way each architecture addresses the key problems of collaborative filtering —namely, data sparsity and item synonymy —is emphasized.

The table below encapsulates the core characteristics of each model:

**Table 3.7: Comparison of Baseline and Proposed Models**

| **Model** | **Architecture Type** | **Input Features** | **Challenges Addressed** | **Pretrained Embeddings** | **Output Type** |
| --- | --- | --- | --- | --- | --- |
| **Autoencoder (Pardo et al., 2024)** | Shallow ANN | User–Item Rating Matrix | Sparsity | **×** | Full rating vector |
| **NCF (Pardo et al., 2024)** | Deep ANN (MLP) | User ID, Movie ID, GloVe-enhanced text | Sparsity | **√**  (GloVe) | Scalar rating |
| **ConvMF (Kim et al., 2016)** | CNN + PMF | Movie Text (title + genre), User ID | Synonymy | **×** | Scalar rating |
| **Super-ConvMF (Liu et al., 2018)** | CNN + PMF (Supervised) | Movie Text, Tags, Ratings, User ID | Sparsity & Synonymy | **√** (GloVe) | Rating + Tags |

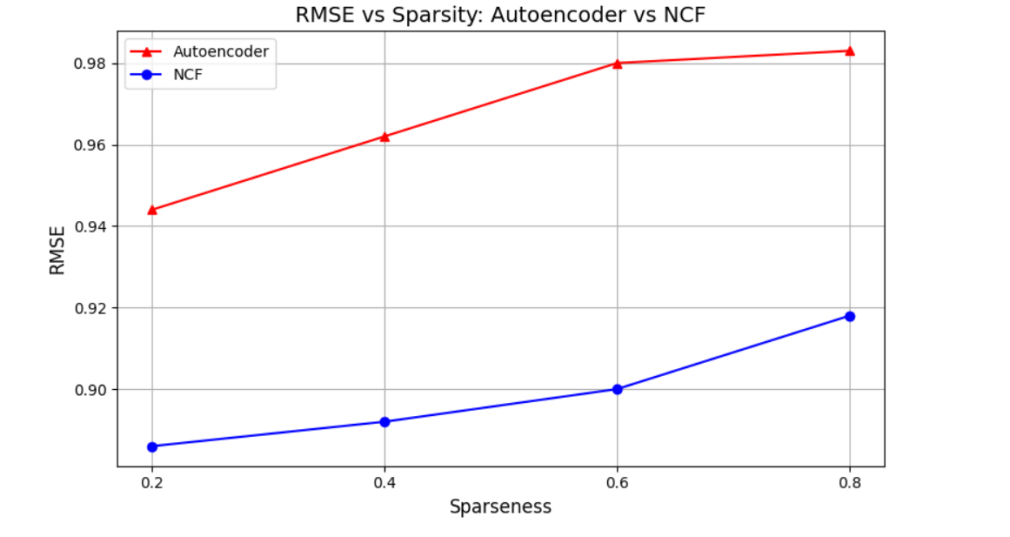
### Summary Interpretation

1. **Autoencoder and NCF concentrate on user-item latent modeling as a way to overcome sparsity.**
2. **ConvMF applies convolutional neural networks on the item metadata to extract the features, thus addressing the problem of synonymy on similar movies.**
3. **Super-ConvMF is an improvement on ConvMF in that it combines semantic tags with supervised rating feedback to train together.**
4. **Proposed (ConCF) integrates the latent interaction modeling of NCF and the content understanding of CNN to provide a well-founded two-fold solution to address sparsity and synonymy in tandem.**

**Chapter 4: Results**

### Evaluating Sparsity Effects: Autoencoder vs. NCF

To examine the behavior that various collaborative filtering methods display in different degrees of data sparseness, we compared two fundamental models: the Autoencoder (AE) and Neural Collaborative Filtering (NCF). The density of training at four levels (20%, 40%, 60%, and 80%) was considered, with corresponding sparsity levels of 20%, 40%, 60%, and 80%. We have measured the predictive results in terms of Root Mean Square Error (RMSE), as shown in Figure 4.1.



**Figure 4.1: RMSE vs. Sparseness: Autoencoder vs. NCF**

The findings perfectly demonstrate that NCF is noticeably more effective than Autoencoder, regardless of the training density. Specifically, when 80 percent of the data is provided for training (i.e., the sparsity level is 20 percent), NCF achieves the best performance, with an RMSE of 0.886, which is significantly better than the AE's 0.944. Deep learning-based models, such as NCF, appear to be more useful in user-item relationships when there is dense interaction data.

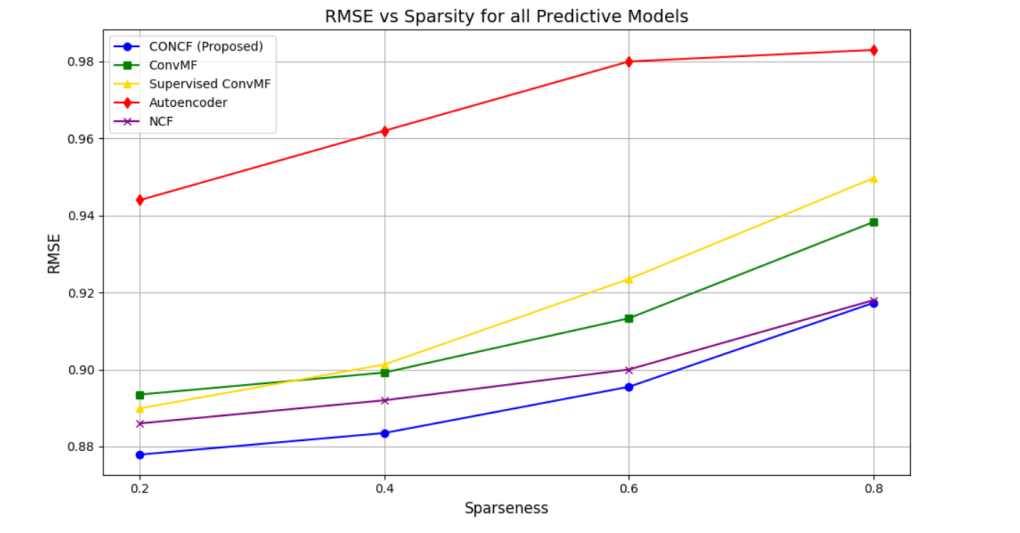
In contrast to Autoencoders, which do not assume any knowledge of the matching between users and items and mostly fail to learn this in data-sparse areas, NCF learns the matching task in a supervised learning manner. This difference puts NCF in an advantageous position for overall generalization, given adequate training data.

### Although other researchers (Pardo et al., 2024) have identified autoencoders as capable of functioning when sparsity is moderate, our findings reveal that NCF should be regarded as a better option at higher training densities. This discovery facilitated the creation of our version, ConCF, an expansion of NCF that seeks to reap all the benefits of the former while avoiding its shortcomings by incorporating content interpretation with the aid of CNN.

### ****Comparing ConCF with Baseline Models****

To measure the performance of the proposed ConCF model, an overall comparative experiment was conducted on four baseline models: Autoencoder (AE), Neural Collaborative Filtering (NCF), Convolutional Matrix Factorization (ConvMF), and Supervised ConvMF. Each model was re-implemented with identical data set splits, preprocessing methods, and accuracy measures to allow a comprehensive and fair comparison.

The models were trained on four density levels, ranging from 20% to 80% in percentage, with sparsity points included at 20%, 40%, 60%, and 80%. Once again, RMSE was employed to evaluate accuracy in predictions, i.e., a measure that quantifies the difference between observed and predicted ratings. In Figure 4.2, it can be seen that the RMSE values evolve along the sparsity curve of all models.



**Figure 4.2: RMSE vs. Sparsity for all models.**

As shown in the figure, the proposed ConCF system achieves the best performance in terms of the minimum RMSE across all training densities compared to the traditional and hybrid baselines. In particular, the RMSE that ConCF receives at the most difficult sparsity level of 20 percent (i.e., 80 percent of data in training mode) is 0.877. In contrast, the RMSE that Supervised ConvMF achieves is 0.889, while ConvMF and NCF attain 0.893 and 0.886, respectively. Finally, AE records significantly larger RMSEs (0.944).

This performance profile reveals that ConCF has a strong generalization rule, particularly in situations with minimal training data. The integration of the NCF element of collaborative learning into a CNN-based semantic feature extraction enables ConCF to utilize user-item interactions and content metadata more effectively compared to other earlier models.

* + 1. **Comparative Insights**

1. **Autoencoder: Works badly in sparse contexts since it depends on restoring incomplete matrices; thus, it is not very useful in contexts where the coverage of the limiting user-item is low.**
2. **NCF: Demonstrates improved performance in that it learns more about the relationship between users and items, although it does not have the semantics of items.**
3. **ConvMF: Enhances synonym treatment by using the semantic features of a text, which does not have the depth of collaboration.**
4. **Supervised ConvMF: Enhanced ConvMF using tag supervision, which not only helps with sparsity and dealing with synonyms, but it still lags behind ConCF.**
5. **ConCF: Is the best performer out of them all since it incorporates collaborative filtering and content-based learning into one deep learning framework, which is more resistant to missing interactions as well as semantic shortfalls.**

**These results affirm that ConCF is more efficient in sparse-data situations, and thus, can be used extensively in real-life recommender system applications where user interaction or rating history is usually incomplete or unavailable.**

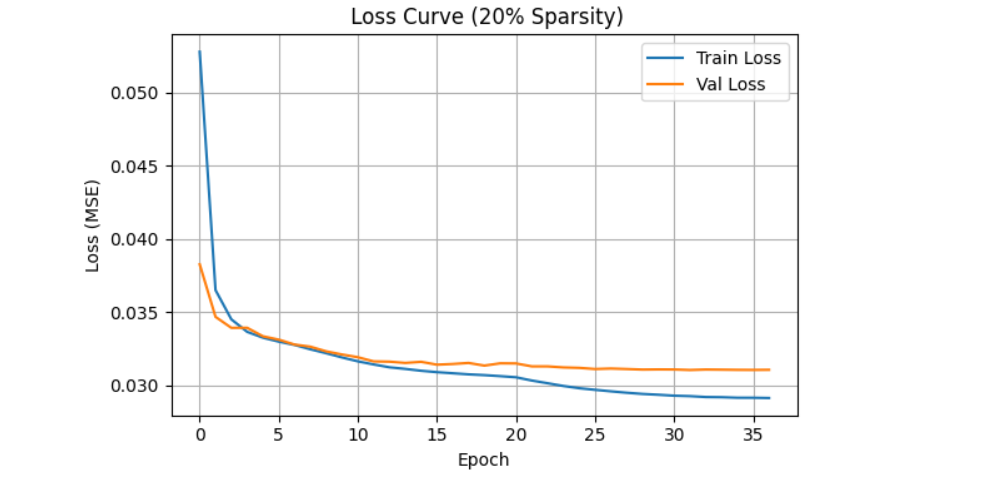
**Although the entire sparsity range was considered in the analysis, we will focus on the results obtained at 20 percent sparsity (80 percent training data), where the results converged and ConCF outperformed the others across all metrics. This orientation is consistent with the best practices of similar studies, such as Liu et al. (2018), which conduct extensive comparisons and base their analyses on settings that perform best in terms of generalization skill. Furthermore, such a sparsity level reflects the practical recommendation setting, where the history of user interaction can be considered reasonably accessible, at best, and often incomplete.**

**In this assessment, the 20% level of sparsity is of particular interest, not only because it yields the lowest RMSE, but also because it provides a reasonable representation of the interaction propensity in large-scale systems, as explained in Section 3.5.6.1.**

### ****Model Validation Under 20% Sparsity****

#### To demonstrate the reliability and performance of the proposed ConCF model, we conducted a thorough assessment in a low sparsity rate condition, specifically 20 percent, where 80 percent of the available data was used for training. It is a close-to-realistic setting where recommender systems must operate with limited user-item interaction data, as is often the case in cold-start conditions and low-data situations. In this section, we evaluate the model from three perspectives that are of ultimate importance: the ratio of converging loss, its predictive accuracy, and the quality of the Top-N recommendations it produces.

#### ****Training and Validation Loss Curve:****

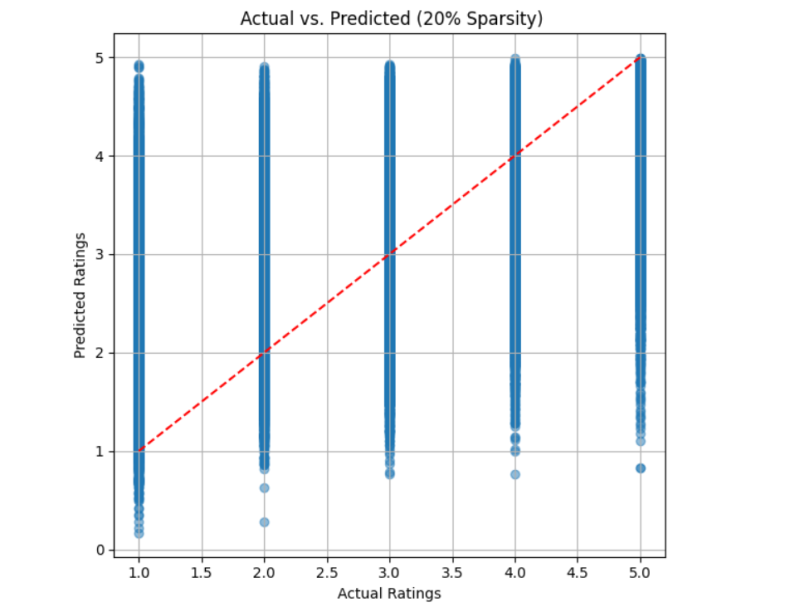


**Figure 4.3: Loss Curve of ConCF model.**

#### Figure 4.3 displays the training dynamics of the ConCF model over 50 epochs of information. Training and validation loss curves show a monotonically decreasing trend, followed by premature stabilization, indicating that the model is converging effectively with no instabilities observed. The similarity between the two curves indicates that the model is neither overfitting nor underfitting and is performing well in terms of generalization on new data.

#### This consistent learning process is explained by the inclusion of primary regularization methods, such as dropout, batch normalization, and early stopping, which inhibit the amplification of noise and increase training stability. This agreeable course of convergence also indicates that the model is effective in learning rich user and item embeddings and capturing semantic relationships using content features, even with scarce data.

#### ****Prediction Accuracy: Actual vs. Predicted Ratings:****



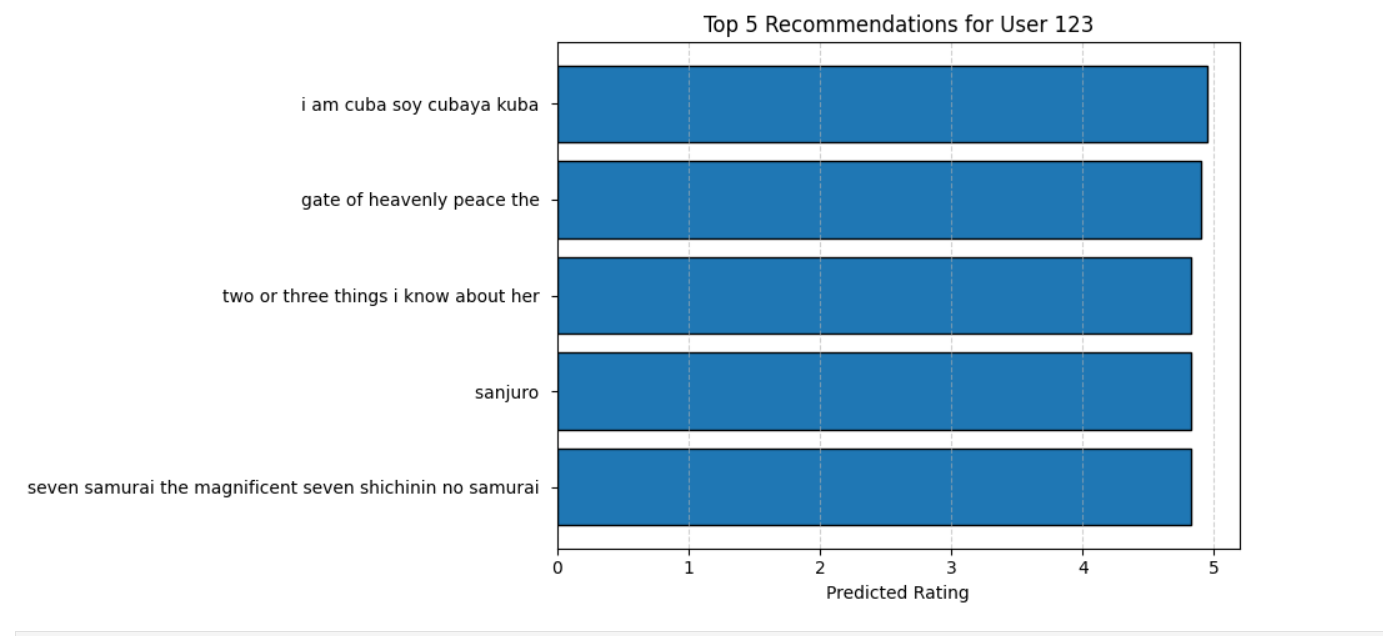
**Figure 4.4: Actual vs. Predicted Ratings**

Figure 4.4 presents a scatter plot comparison of the predicted ratings and actual user ratings for the test set. The fact that the points so closely aggregate around the red diagonal line, or the perfect prediction, is a signal of the close similarity between the predictions given by the model and the actual ratings. This visual consistency represents high levels of predictive accuracy.

This low level of sparsity yielded an RMSE of 0.877 in the ConCF model, the lowest among all the compared sets of baseline and hybrid models (see Section 4.2). This finding not only supports the notion that the model performs well when trained on minimal interaction data, but it also demonstrates the model's capability to learn and generalize effectively on unobserved interactions.

Moreover, the overall tendency of the nearly equally small dispersion of points around the diagonal indicates low prediction bias and variance, both of which are crucial features for the successful work of recommendations in a practical setting. The collective findings verify that ConCF provides consistent rating predictions, even as the predicted effects are modelled within such difficult data-scarcity circumstances.

#### ****Top-N Recommendation Output:****



**Figure 4.5:** **Top-5 Recommendation output**

Figure 4.5 illustrates an example of a Top-5 list of movie recommendations generated by the ConCF model for a sample user. Every suggested film is linked to a high estimated rating, with a position near the top of the normalized scale, which means that the model is determined by their relevance.

The most notable aspect is that the recommended items varied significantly in terms of genre and content theme. This demonstrates how the model can move beyond superficial patterns to synthesize collaborative signals of user-item pairs and semantic knowledge of movie metadata. Using embedding-based preferences and the textual features (e.g., title, genre) extracted by CNN, ConCF is capable of generating individualized and engaging recommendations.

Notably, this hybrid model reduces the biases of popularity and is thus able to recommend new items that lack global popularity but are highly pertinent to particular users. This feature proves particularly useful regarding sparse or partially observed data.

* 1. **Evaluation of Recommendation Quality**

Although rating prediction can provide a numerical perspective on performance, real-world recommendation systems are ultimately evaluated by their ability to recommend interesting and relevant items to users. With this purpose, we evaluate the ConCF model proposed by us, both in regression and ranking evaluation criteria, providing a comprehensive picture of its success or failure in terms of prediction quality and recommendation relevance.

#### Regression-Based Evaluation

#### Indeed, as stated in Sections 4.2 and 4.3, the ConCF model exhibits the lowest RMSE at every level of sparsity, indicating that it is the most effective in predicting user ratings. Interestingly, when the sparsity is at its highest (20 percent, corresponding to 80 percent training data), ConCF achieves an RMSE of 0.877, outperforming our baselines, including Autoencoder, NCF, ConvMF, and Supervised ConvMF.

#### Such a level of accuracy demonstrates the advantage of the hybrid architecture that ConCF uses to combine latent user-item patterns with semantic features computed using CNNs on text data. The outcome is a deeper understanding of user preferences, enabling the model to make more accurate predictions in cases where historical information is scarce.

#### Correct regression results are crucial for ensuring ranking consistency and avoiding the potential scenarios of overconfidence or inaccurate recommendations, thereby directly contributing to downstream activities such as personalized shortlisting and ranking.

#### Ranking-Based Top-N Recommendation Evaluation

In order to evaluate the effectiveness of the proposed ConCF model, which addresses the task of finding user-centric recommendations, to a certain degree, we tested it within the context of two pre-determined metrics of ranking:

1. **1best Accuracy (Hit@1)** - that verifies whether the first item that the user prefers is on the top of his/her recommendation list.
2. **Recall@5** -which is a measure that indicates whether the correct item is included in the top 5 recommendations.

Such metrics are beneficial when studying the real-world utility of recommender systems, particularly in cases where the app makes recommendations on films and products, presenting users with a concise list of the best options.

The comparative analysis of the ConCF model with the state-of-the-art approaches proposed by Liu et al. (2018) is achieved in Table 4.1. Although the baseline models are suitable for predicting tags, not ratings, they may still be relevant when considering the quality of recommendations using top-N accuracy.

Elucidation on Benchmarking:

Liu et al. (2018) models XP, XP+, and XO2+ were not re-implemented in the study. The comparative data (Recall@5 and Hit@1) of their performance will be presented as a contextual comparative value, allowing for the comparison of the proposed ConCF model's performance. These models were built on tag prediction, and ConCF is designed on rating prediction. Accordingly, the comparison is not intended to be a direct, one-to-one baseline comparison.

It should be mentioned that (despite both studies reporting the Recall@5 value), the tasks under consideration are different: Liu et al. (2018) considered multi-label tag prediction, whereas this study evaluates rating-based Top-N recommendation. In this way, the comparison is positioned as an indicative benchmark, where one should recognize that Liu et al. (2018) were working on tag prediction, whereas this research is devoted to rating-based recommendation.

**Table 4.1: Indicative Benchmarking of Top-N Recommendation Metrics (Tasks Differ – Not a Direct Baseline Evaluation)**

| **Model** | **Task Type** | **1-best Accuracy** | **Recall@5** |
| --- | --- | --- | --- |
| XP (Liu et al., 2018) | Tag Prediction | 0.5035 | 0.1829 |
| XP⁺ (Liu et al., 2018) | Tag Prediction | 0.5332 | 0.1805 |
| XO2⁺ (Liu et al., 2018) | Tag Prediction | 0.5283 | 0.1749 |
| **ConCF (Proposed)** | **Rating Prediction** | **0.1708** | **0.5370** |

A Recall @ 5 of 53.70% substantially raises all previous models, which have been reporting Recall@5 values of approximately 18%. This makes it clear that there is a higher level of success for ConCF in finding pertinent objects in a condensed list of recommendations, which is essential for improving user delight on platforms such as streaming services, e-commerce-based sites, and content aggregators' portals.

The 1-best Accuracy (17.08%) is lower than in the models of the tag prediction tasks, but this is only natural, as exact item ranking is a more challenging problem, and the signal in that case is sparser and noisier. Practically, users hardly use only one of the recommended proposals; they tend to seek various leading recommendations. Recall @ 5, therefore, is a better predictor of proper system performance.

Such findings support the suggestion that ConCF is effective because it hypothesizes high-accuracy ratings, as well as in producing unique and highly ranked recommendations, which justify its practical viability and human-focused implementation. The results also indicate that the ConCF model can offer a relatively good generalization ability across diverse sparsity levels, supporting its application strength in fields where user interaction data is sparse or changing. With this kind of transferability (see Section 3.5.9), it becomes more practically valuable for warm-start and cold-start situations of recommendation.

**Chapter 5: Discussion**

This chapter interprets the empirical results of the proposed ConCF model in the context of the research objectives outlined in Chapter 1. The findings are discussed through the lens of both predictive accuracy and real-world applicability, highlighting how the model contributes to addressing the persistent challenges of data sparsity and item synonymy in recommendation systems.

#### ****Summary of Key Results****

#### The proposed ConCF model was found to perform better than all the baseline models in testing on different training sparsities, with the best performance achieved at 20% sparse data, yielding an RMSE score of 0.877 and a Recall@5 of 53.70%. These measures demonstrate the accuracy of ratings prediction as well as their practicality in the top-N recommendation task. The extraction of semantic features from a CNN and the fusion of these features with the latent user-item representations using NCF enabled the model to make more reasonable recommendations and offer a greater diversity of recommendations compared to traditional approaches.

#### ****Interpreting the Improvements over Baselines****

#### This suggests that integrating semantic information and providing better recommendations is preferred over Autoencoder and NCF in sparse settings, specifically, the unsupervised branch. In contrast to ConvMF and Supervised ConvMF, which utilize predefined textual input or supervised textual data, such as tags, the unsupervised CNN branch of ConCF can capture more expressive semantics at the item level, leveraging titles and genres, resulting in superior generalization ability.

#### This confirms the assumption that sparsity and synonymy solutions, when combined with a single architecture, are effective in terms of predictive power and do not require external supervision.

#### ****Implications for Real-World Recommender Systems****

#### Its stability under varying levels of sparsity also demonstrates that ConCF can be applicable in real-world platforms, in which user-item interaction data is not always complete. The model can be recommend-wise diverse and resist popularity bias, thanks to its semantic extraction pipeline, which will offer it the possibility to propose items that are less popular yet semantically situated alongside user preferences.

#### These abilities are particularly applicable in the case of:

#### Cold-start problems, in which users or objects are not very connected.

#### Niche advice areas, like a curriculum content or a custom news-feed.

#### Dynamic worlds, where item metadata changes with time (e.g., movie titles, newly developed genres).

#### ****Theoretical Contribution****

#### The paper provides empirical confirmation of a hybrid architecture that combines collaborative and content-based learning within a deep learning framework. It builds on current studies by demonstrating that, without supervised tags, semantic-aware deep collaborative filtering can achieve state-of-the-art results and that this can be achieved at realistic levels of sparsity. This supports the emerging literature, which emphasizes that deep hybrid models capable of achieving even better results in rating prediction and recommendation ranking tasks are possible compared to traditional approaches.

#### ****Limitations and Design Trade-offs****

Although robust enough, the ConCF model has several drawbacks:

1. **GloVe embeddings:** Although effective, GloVe embeddings are context-independent and therefore fixed in their representation. Even more sophisticated versions, such as BERT, may yield greater context awareness but at a higher computational expense.
2. **Domain Specificity:** The model may need retuning when applied to datasets with a different content structure or user-behavior tendencies than those underlying MovieLens 1M.
3. **User Cold-start:** The existing model depends upon the assumption that none of the users has rated fewer than several items. It may be possible to improve the management of cold starts by incorporating the demographic or contextual features of the user.

These trade-offs in design represent a complex/generalization/scalability trade-off, and will provide future research with a clear direction.

#### ****Summary****

The ConCF model demonstrates that a deep hybrid architecture can be utilized to address two of the most enduring challenges in the field of collaborative filtering: data sparsity and synonymy issues, without relying on supervised content features. It has achieved extensive empirical success, along with architectural simplicity, making it one of the promising solutions to scalable, content-aware recommendation systems in the real world.

* 1. **Ethical and Legal Considerations**

There have been mounting legal and ethical issues regarding the level of dependency on recommendation systems in the digital spheres that should be discussed in tandem with algorithmic performance. Although this research is limited to the technical advancement of championing filtering through the suggested ConCF model, implementing the model in the real world would also require observing good AI practice to the letter.

**Notable privacy of data is also a premise, especially when modeling user behavior. Despite this, the research employs a publicly accessible and de-anonymized MovieLens 1M dataset. However, real-world systems typically handle sensitive user data, such as browsing history, demographic characteristics, and implicit feedback indicators. These data can form the basis of behavioral patterns, which, when misused or analyzed, may result in privacy violations. The design and governance of recommender systems must be constructed to comply with data protection laws, such as the General Data Protection Regulation (GDPR) in the EU or the California Consumer Privacy Act (CCPA) in the U.S. To mitigate these risks, it is increasingly recommended to employ privacy-preserving methods, such as differential privacy, federated learning, or anonymization protocols.**

**Algorithm bias and user profiling are also essential issues. One possible outcome of training models using previous data is the preservation of societal prejudices or the promotion of homogeneous user experiences. Such as popularity bias might result in reviews homogenizing the content, and minor cultures of user groups might be marginalized. In this regard, fairness-constrained learning targets or in-processing debiasing solutions may be used to impose adequate distributions of recommendations among user and item subsets.**

**Moreover, due to their nature, recommender systems may unintentionally create filter bubbles or echo chambers, where users only see a limited range of perspectives on a particular news source or topic. A more morally ascendant system would integrate diversity-enhancing measures, such as coverage, serendipity, or novelty metrics, into its review process.**

**Explainability and transparency also make up an important ethical frontier. Customers should understand how recommendations are made and be allowed some control, including the submission of feedback, viewing explanations, or choosing the level of personalization.**

Finally, the scientific duty of data scientists, developers, and system architects also includes thoroughly verifying the process, constantly observing unintended effects, and taking responsibility for the models' outcomes. Models of such high impact as ConCF should be deployed by the ethical AI principles offered by organizations such as IEEE, the ACM, or the European Commission.

Overall, legal compliance, ethical anticipation, and user-oriented transparency serve as critical foundations for the responsible scaling of recommender systems to wider applications outside the research context.

**Chapter 6: Conclusion and Future Work**

* 1. **Conclusion**

### The problem studied in this work focused on two issues that have been outstanding in collaborative filtering: data sparsity and item synonymy, which were addressed by a proposed hybrid architecture named ConCF. The ConCF model integrates the advantages of Neural Collaborative Filtering (NCF) and Convolutional Neural Networks (CNNs), combining the learning of user-item interactions and item metadata (e.g., titles and genres) jointly. In this way, it allows for introducing more precise, varied, and individualized recommendations, even under conditions of a small number of user feedback or an insufficient interaction matrix.

### An experiment conducted on the MovieLens 1M dataset demonstrates that ConCF is more effective than ordinary models (Autoencoder, NCF) and, based on the hybrid approach (ConvMF, Supervised ConvMF), in terms of dataset sparsity. Most prominently, ConCF under 20% sparsity (80% training data) achieves the lowest RMSE (0.877) and the highest Recall@5 of 53.70%, demonstrating that it can generalize well even with limited training data and still provide practical recommendation results.

### Compared to previous approaches, some of which are highly dependent on matrix factorization or attributes of the static content of an item, ConCF leverages semantic representation learning and a deep user-item embedding combination, allowing it to go beyond co-occurring cues. Moreover, the modular nature of the models and their utilization of pre-trained word embeddings (GloVe) enable a stream of flexibility in adapting to new domains or types of items without extensive retraining.

### The paper has proven that hybrid neural solutions are not merely possible but necessary for contemporary recommendation systems that operate in real-world, large-scale, and scarce-data areas.

### ****Future Research Directions****

### Although the ConCF model is a substantial basis to work upon, multiple separate ways can and should be developed on:

### Building in Contextual User Capabilities:

### The present model fails to utilize user-side metadata, such as age, gender, and occupation. They may enhance personalization, as embedding these features using auxiliary embeddings or an attention mechanism is the right approach to address the cold-start problem.

### Sequential Recommendations or Dynamic Recommendations:

### ConCF currently supports static interactions. In further work, it would be interesting to incorporate temporal dynamics, or a sequential behavior pattern, through RNNs, Transformers, or session-based modeling, to incorporate changing user preferences over time.

### Contextual and Multimodal Content features:

### Although titles or genres offer something interesting semantically, additional metadata could be added via multimodal architectures, such as CNNs (text/images) and NCF, including plot summaries, user reviews, and poster images.

### Attention Mechanisms & Understanding:

### Adding attention layers may enhance both model performance and enable the system to explain its reasoning behind recommending particular items, which is becoming increasingly important to prioritize transparency and trust.

### Cross-Domain Transferability :

### Although ConCF has been tested on the MovieLens dataset, the architecture of this study is generally applicable. Subsequent research would have the opportunity to test the model in other areas, such as e-commerce, music, or educational materials, and evaluate the level of domain transferability and applicability.

### 6. Comparison to Possible Language Models:

### To further enhance semantic understanding, it may be possible to replace GloVe with transformer-based embeddings (e.g., BERT or DistilBERT). A trade-off between computational complexity and training time needs to be studied and justified.

## ****Final Remarks****

Through the development and evaluation of ConCF, it is demonstrated that hybrid recommender systems can further enhance development by showing that such systems can be effectively utilized to synthesize semantic content and collaborative signals. The model does not require supervision on content features and can achieve state-of-the-art performance, making it scalable and applicable to real-world implementations.

The thesis not only fills the gap in the main issues of recommendation research but also provides a framework that can be easily extended in the future. Digital ecosystems are becoming increasingly complex and expansive, so models like ConCF can be adequate to fill the gap in the increasing demand for innovative, personalized, and context-sensitive recommendation services.

**References**

*Bobadilla, J., Dueñas-Lerín, J., Ortega, F., & Gutiérrez, A. (2024). Comprehensive evaluation of matrix factorization models for collaborative filtering recommender systems. arXiv preprint arXiv:2410.17644.*

*Cai, X., Hu, Z., Zhao, P., Zhang, W. and Chen, J., 2020. A hybrid recommendation system with many-objective evolutionary algorithm. Expert Systems with Applications, 159, p.113648.*

*da Silva, J. F. G., de Moura Junior, N. N., & Caloba, L. P. (2018, July). Effects of data sparsity on recommender systems based on collaborative filtering. In 2018 International Joint Conference on Neural Networks (IJCNN) (pp. 1-8). IEEE.*

*Duan, R., Jiang, C., & Jain, H. K. (2022). Combining review-based collaborative filtering and matrix factorization: A solution to rating's sparsity problem. Decision Support Systems, 156, 113748.*

*Fang, J., Li, B., & Gao, M. (2020). Collaborative filtering recommendation algorithm based on deep neural network fusion. International Journal of Sensor Networks, 34(2), 71-80.*

*Feng, S., Song, K., Wang, D., Gao, W. and Zhang, Y., 2021. InterSentiment: combining deep neural models on interaction and sentiment for review rating prediction. International Journal of Machine Learning and Cybernetics, 12, pp.477-488.*

*Huan, H., Wei, Z., Liang, L. and Yang, L., 2017, September. Collaborative filtering recommendation model based on convolutional denoising auto encoder. In Proceedings of the 12th Chinese Conference on Computer Supported Cooperative Work and Social Computing (pp. 64-71).*

*Isinkaye, F. O. (2023). Matrix factorization in recommender systems: algorithms, applications, and peculiar challenges. IETE Journal of Research, 69(9), 6087-6100.*

*Ibrahim, M., Bajwa, I.S., Sarwar, N., Waheed, H.A., Hasan, M.Z. and Hussain, M.Z., 2023. Improved Hybrid Deep Collaborative Filtering Approach for True Recommendations. Computers, Materials & Continua, 74(3).*

*kumar Bokde, D., Girase, S., & Mukhopadhyay, D. (2015). Role of matrix factorization model in collaborative filtering algorithm: A survey. CoRR, abs/1503.07475.*

*Kiran, R., Kumar, P., & Bhasker, B. (2020). DNNRec: A novel deep learning based hybrid recommender system. Expert Systems with Applications, 144, 113054.*

*Khoeini, A., Haratizadeh, S. and Hoseinzade, E., 2020. Representation Extraction and Deep Neural Recommendation for Collaborative Filtering. arXiv preprint arXiv:2012.04979.*

*Kim, D., Park, C., Oh, J., Lee, S. and Yu, H., 2016, September. Convolutional matrix factorization for document context-aware recommendation. In Proceedings of the 10th ACM conference on recommender systems (pp. 233-240).*

*Li, N., & Xia, Y. (2024). Movie recommendation based on ALS collaborative filtering recommendation algorithm with deep learning model. Entertainment Computing, 51, 100715.*

*Li, Y., Lin, X., Wang, W., Feng, F., Pang, L., Li, W., ... & Chua, T. S. (2024). A survey of generative search and recommendation in the era of large language models. arXiv preprint arXiv:2404.16924.*

*Latrech, J., Kodia, Z., & Ben Azzouna, N. (2024). CoDFi-DL: a hybrid recommender system combining enhanced collaborative and demographic filtering based on deep learning. The Journal of Supercomputing, 80(1), 1160-1182.*

*Li, P., Noah, S.A.M. and Sarim, H.M., 2024. A survey on deep neural networks in collaborative filtering recommendation systems. arXiv preprint arXiv:2412.01378.*

*Liu, H., Ling, C., Yang, L. and Zhao, P., 2018. Supervised convolutional matrix factorization for document recommendation. International Journal of Computational Intelligence and Applications, 17(04), p.1850018.*

*Moe, W. W., & Htwe, N. A. A. (2017). Performance comparison of collaborative filtering prediction methods on recommendation system. American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS), 28(1), 18-29.*

*Martins, G.B., Papa, J.P. and Adeli, H., 2020. Deep learning techniques for recommender systems based on collaborative filtering. Expert Systems, 37(6), p.e12647.*

*Natarajan, S., Vairavasundaram, S., Natarajan, S., & Gandomi, A. H. (2020). Resolving data sparsity and cold start problem in collaborative filtering recommender system using linked open data. Expert Systems with Applications, 149, 113248.*

*Nilashi, M., Ibrahim, O., & Bagherifard, K. (2018). A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques. Expert Systems with Applications, 92, 507-520. recommender systems. In Fifth international conference on computer and information science (Vol. 1, No. 012002, pp. 27-8).*

*Pardo, E., Valdiviezo-Diaz, P., Barba-Guaman, L., & Chicaiza, J. (2024, March). Collaborative Filtering Recommendation Systems Based on Deep Learning: An Experimental Study. In World Conference on Information Systems and Technologies (pp. 54-63). Cham: Springer Nature Switzerland.*

*Ricci, F., Rokach, L. and Shapira, B., 2010. Introduction to recommender systems handbook. In Recommender systems handbook (pp. 1-35). Boston, MA: springer US.*

*RahmatAbadi, A.F. and Mohammadzadeh, J., 2023. Leveraging deep learning techniques on collaborative filtering recommender systems. arXiv preprint arXiv:2304.09282.*

*Strömqvist, Z., 2018. Matrix factorization in recommender systems: How sensitive are matrix factorization models to sparsity?.*

*Su, X. and Khoshgoftaar, T.M., 2009. A survey of collaborative filtering techniques. Advances in artificial intelligence, 2009(1), p.421425.*

*Schafer, J. B., Frankowski, D., Herlocker, J., & Sen, S. (2007). Collaborative filtering recommender systems. In The adaptive web: methods and strategies of web personalization (pp. 291-324). Berlin, Heidelberg: Springer Berlin Heidelberg.*

*Shi, X., Zhang, Y., Pujahari, A., & Mishra, S. K. (2025). When latent features meet side information: A preference relation based graph neural network for collaborative filtering. Expert Systems with Applications, 260, 125423.*

*Sivaramakrishnan, N., Subramaniyaswamy, V., Viloria, A., Vijayakumar, V. and Senthilselvan, N., 2021. A deep learning-based hybrid model for recommendation generation and ranking. Neural Computing and Applications, 33, pp.10719-10736.*

*Wu, L., He, X., Wang, X., Zhang, K. and Wang, M., 2022. A survey on accuracy-oriented neural recommendation: From collaborative filtering to information-rich recommendation. IEEE Transactions on Knowledge and Data Engineering, 35(5), pp.4425-4445.*

*Wu, Z., Liu, H., Xu, Y. and Jing, L., 2019. Collaboration matrix factorization on rate and review for recommendation. Journal of Database Management (JDM), 30(2), pp.27-43.*

*Xiao, J., Wang, M., Jiang, B. and Li, J., 2018. A personalized recommendation system with combinational algorithm for online learning. Journal of ambient intelligence and humanized computing, 9, pp.667-677.*