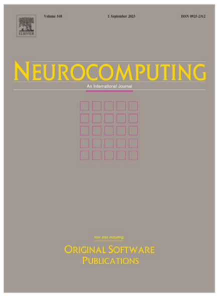
Journal Pre-proof



A collaborative filtering recommender systems: Survey

Mohammed Fadhel Aljunid, Manjaiah D.H., Mohammad Kazim Hooshmand, Wasim A. Ali, Amrithkala M. Shetty, Sadiq Qaid Alzoubah

PII: S0925-2312(24)01489-9
DOI: https://doi.org/10.1016/j.neucom.2024.128718
Reference: NEUCOM 128718

To appear in: *Neurocomputing*

Received date : 1 July 2022
Revised date : 16 September 2024
Accepted date : 12 October 2024

Please cite this article as: M.F. Aljunid, Manjaiah D.H., M.K. Hooshmand et al., A collaborative filtering recommender systems: Survey, *Neurocomputing* (2024), doi: https://doi.org/10.1016/j.neucom.2024.128718.

A Collaborative Filtering Recommender Systems: Survey

Mohammed Fadhel Aljunid, Manjaiah D.H, Mohammad Kazim Hooshmand, Wasim A. Ali, Amrithkala M Shetty, Sadiq Qaid Alzoubah[a,c,1,1,1,1]

[a] *Dept. of Computer Science, College of computers & Informatics, Thamar University , Thamar, Yemen*
[b] *Dept. of Computer Science, Mangalore University, konaje, Mangalore, 574199, Karnataka, India*
[c] *Dept. of Computer Science, Mangalore University, konaje, Mangalore, 574199, Karnataka, India*
[d] *Department of Computer Science and Engineering, PG Centre, Hanchya Sathagally Layout, Mysore, 570019, Karnataka, India*
[e] *Dep of Computer Science, Mangalore University, konaje, Mangalore, 574199, Karnataka, India*
[f] *Dept. of Computer Science, Mangalore University, konaje, Mangalore, 574199, Karnataka, India*

Abstract

In the current digital landscape, both information consumers and producers encounter numerous challenges, underscoring the importance of recommender systems (RS) as a vital tool. Among various RS techniques, collaborative filtering (CF) has emerged as a highly effective method for suggesting products and services. However, traditional CF methods face significant obstacles in the era of big data, including issues related to data sparsity, accuracy, cold start problems, and high dimensionality. This paper offers a comprehensive survey of CF-based RS enhanced by machine learning (ML) and deep learning (DL) algorithms. It aims to serve as a valuable resource for both novice and experienced researchers in the field of RS. The survey is structured into two main sections: the first elucidates the fundamental concepts of RS, while the second delves into solutions for CF-based RS challenges, examining the specific tasks addressed by various studies, as well as the metrics and datasets employed.

Keywords: Survey, Recommender System, Collaborative Filtering, Machine Learning, Deep Learning, Data Sparsity, Accuracy, Cold start

1. Introduction

Recommender systems (RSs) enable users to discover items or services that are relevant to their needs or interests, based on a variety of data sources, such as historical behavior, social relationships, points of interest, and other user data. In addition to explicitly collected user ratings, implicit data can also be gathered by monitoring user behavior, such as songs listened to, applications downloaded, websites visited, books read, and so on. The primary goal of an RS is to generate personalized product or service recommendations that are most likely to be of interest to the user. RSs are ubiquitous, with examples ranging from Amazon and Netflix to Facebook and LinkedIn. By presenting tailored suggestions, RSs effectively narrow down the vast array of options available to users.

The formulation of recommendation problems is primarily intended to address one or both of two fundamental questions: the Predicted version and the Ranking version. The Predicted version involves predicting evaluation values for user-product interactions. In this scenario, the training data comprises evaluation values provided by users, and the goal is to use this data to predict the evaluation value of items with which the user has not yet interacted. On the other hand, the Ranking version focuses less on predicting user behavior and more on presenting a limited list of the best items for users to consider. This is particularly relevant for retailers and e-commerce companies that prioritize the creation of curated lists of high-quality products. From the user's perspective, the ability of the system to predict their rating for a specific product may be less important than the system's ability to recommend items they are likely to enjoy.

In recent years, RS has become more widely implemented on the Internet, which has made it easier to use in a variety of contexts Research papers on movie recommendation studies make up the majority of the literature [1, 2]; however, a great volume of literature for RS is centered on different topics, such as music [3, 4, 5], television [6, 7], books [8, 9], documents [10, 11, 12, 13], e-learning [14, 15], e-commerce [16, 17], applications in markets [18] and web search[19], among others.

The main objective of RS is to increase the company's sales. All the system has to do for this purpose is to show or provide products that are meaningful to the user. They try to balance factors like relevance, novelty, serendipity, and diversity. The kinds of filtering most used of the RS are collaborative [20, 21, 22], content-based[23, 24], and Hybrid RS [25, 20].

In the suggestion, algorithms of CF play an essential part. The CF algorithm is founded on the premise that a decent approach for a user to discover the information they are interested in using is first to find other users who share similar interests with this user. The fundamental concept is straightforward. CF algorithm is also frequently used in conjunction with other filtering techniques, such as content-based, knowledge-based, or social ones. Most of the decisions we make daily are based on the recommendations of people we trust. CF applies this concept to its e-commerce RS, which recommends targeting users based on other users' evaluations of specific content. Although CF has many of the same applications as a typical recommendation technology, there are still many issues that need to be addressed. These are the most common problems: a cold start, sparsity, and scalability [26, 27].

Content-based is defined by the relevant content's characteristic attributes in the content-based RS. Based on the characteristics of the user's evaluation object, the system learns the user's interest and examines the degree to which the user's profile matches the item to be predicted. The learning method used determines the user's profile model [28, 29, 30]. Decision trees, neural networks, and vector-based representation methods are commonly used. A content-based user profile necessitates the user's historical data, and the user profile model may change as the user's preferences change. The benefits include no cold start, new items, data-sparse issues, and the ability to recommend users with specific hobbies. While there are some drawbacks, the recommended content should be able to abstract meaningful features and have a good structure.

Hybrid techniques typically combine various methodologies to get better recommendation outcomes. Recommendations on modern websites usually do not rely entirely on a single RS and technique [31, 32, 33].

As the field of CF algorithms was being developed, several survey papers were published that summarized the most significant challenges in the domain. To provide a concise overview, the authors of these papers have selected the most relevant aspects of CF to comprehend its evolution. However, current surveys tend to be more general, whereas our survey aims to enhance the evolution of CF RS specifically, from the traditional methods of the first phase to the machine learning (ML) techniques of the second phase, and ultimately to the deep learning (DL) methods of the third phase.

To cater to new readers, we have included some conventional themes in this survey, such as CF foundations, limits, and assessment measures. Nonetheless, the crux of our study is to focus on unique issues that have not been covered in previous surveys. We aim to provide a comprehensive understanding of the evolution of CF RS, enabling researchers and practitioners to leverage the advancements in machine learning and deep learning to improve the accuracy and effectiveness of recommendation systems. Through this review, advanced CF readers will explore in depth the most common CF solutions, problems, datasets, and techniques. Readers interested in brand-new and future applications will benefit from this study since it provides information on the most current developments in CF RS trends.

The structure of this work is organized as follows: The related works are given in Section 2, a methodology is presented in Section 3, the CF algorithm foundations are given in Section 4, Remedies for challenges of CF based on ML and DL approaches in Section 5, An evaluation of comprehensive literature in quantitative terms in Section 6, Evolutionary Journey of CF in Section 7 of CF and finally, the conclusion section is given in Section 8.

2. Related Works

While collaborative filtering (CF) algorithms have grown in popularity and complexity, several survey articles have been published in this field. In [34], the authors highlighted key elements of the CF, such as theory and practice, assessment, interactions, and the protection of privacy. In [35], the authors provided a survey of the main concept of CF algorithms and compared their results. In [36], the authors presented a review of CF algorithms. The authors introduce the concept of CF and address the main deficiencies of CF in a concise manner, such as sparsity, scalability, grey sheep, synonymy, shilling attacks, privacy, and so on. They also provided a table with an overview of CF algorithms. In [37], the authors collect a collection of 210 RS publications on RS surveyed and categorize them according to the year and the journal published, their application domain, and their data mining algorithms. They also divided the articles into eight application domains, such as movies, music, etc. In [38], the authors gave an overview of RS algorithms. They concentrated on explaining in detail how the most commonly used RS techniques work. They also discussed the fundamental hypothesis of CF, in addition to their dimensionality reduction algorithms, evaluation metrics, diffusion-based models, social filtering, and metamethods. In [39], the authors gave an overview of CF RS. They also described their evolution, provided an original classification for these algorithms, identified future implementation domains, and developed specific areas chosen for past, present, or future importance. In [20], the authors presented a survey of CF algorithms, ranging from conventional algorithms to social network-based blend algorithms. In [24], the authors presented a CF algorithm survey. They were only interested in active learning models in CF RS.

It is worth noting that while the studies mentioned above have provided valuable insights into the design and implementation of different types of RSs, they have not specifically addressed the role of new technologies, such as DL, in addressing the challenges associated with the CF algorithm. Given the increasing importance of DL in various domains, there is a growing need to explore its potential in the context of RSs. Our proposed work aims to fill this gap by providing guidelines for leveraging the power of ML and DL algorithms in building more accurate and effective RSs.

In [40], the authors presented a review and discussion of existing deep RS methods. However, because this review study only includes a small number of publications, it provides a very limited perspective on the entire concept.

In [41], the authors reviewed traditional RS in addition to DL methods. This review is also inappropriate for its topic because it only examines three publications. In [42], the authors conducted a review of DL recommendation models and created a categorization framework for categorizing processes based on input and output properties. They only gave a few details about the survey. In [43], the authors provided a thorough examination of DL for RS. Even though the number of reviewed papers in this study is very close, the category methods differ.

In [25], the authors have released a survey on DL for RS, including problems and solutions. In [44], the authors have provided a survey with a systemic summary of three types of RS: trust-aware RS, social-aware RS, and robust RS. All of the existing studies are, in general, limited papers, and there is no study dedicated to DL for CF, which is regarded as the most important algorithm for building an RS.

While the above-mentioned studies provide valuable insights into the current state of the field, it is important to note that the review only includes a small number of publications. As a result, the studies may not provide a comprehensive perspective on the entire concept of deep RS methods. Furthermore, the review is limited to a specific period and may not account for more recent developments in the field. In addition, the authors did not evaluate the performance of the deep CF RS methods discussed in the study, which may limit the usefulness of their findings for practitioners interested in applying these methods in real-world settings. Future research could address these limitations by conducting a more comprehensive review of the literature and by evaluating the performance of deep RS methods using real-world datasets.

Our proposed work addresses a crucial gap in the existing literature by focusing on the application of ML and DL algorithms in building CF-based RS. We conduct a comprehensive review of the existing literature on traditional and automated CF-based RS, with a specific focus on ML and DL approaches. While there has been extensive research on CF-based RS, there is a lack of studies that specifically examine the use of ML and DL techniques in this context. Therefore, Our goal is to provide a broad understanding of the subject and to highlight the key challenges and opportunities associated with using ML and DL algorithms in CF-based RS. While we do not delve into implementation details, we provide a detailed analysis of the solutions, tasks, and datasets that have been used in the development of ML and DL models for CF-based RSs. Our study aims to facilitate future research in this area and encourage the adoption of advanced ML and DL techniques for building more accurate and effective CF-based RSs.

3. Methodology

In this study, we adopted a scientific and rigorous approach to selecting research papers related to collaborative filtering (CF)-based recommender systems (RS) algorithms. We initially compiled a comprehensive collection of approximately 800 research papers. To ensure the thoroughness of our survey, we meticulously selected 320 systematic references through a stringent and comprehensive process. Our literature search encompassed multiple databases and search engines, including Google Scholar, Scopus, and IEEE Xplore, utilizing a combination of keywords and filters to refine the search results to the most relevant and recent publications in the CF RS domain. Each potential reference was evaluated based on several criteria, including the quality of the research, the significance of the publication venue, and its relevance to the topic of the survey. Additionally, we considered the number of citations and the impact factor of the publication venue as indicators of the reference's importance and influence in the field. Following this initial evaluation, we further narrowed down the list of references through a detailed analysis of the content and methodology of each publication. We prioritized references that offered novel insights, robust research designs, and reliable data while excluding those based on outdated or unreliable information.

Our meticulous selection process and rigorous approach ensured that our survey is grounded in the most current and reliable information available, providing a comprehensive and unbiased overview of the state of CF-based RS. This study offers a thorough understanding of CF-based RS, including an in-depth explanation of the fundamental concepts of CF-based RS, with a particular emphasis on the CF algorithm. We explore key concepts, types of publications, popular filtering algorithms, evaluation metrics, and the limitations of CF-based RS. Additionally, we provide insights into emerging solutions and trends within the CF RS field, making our survey a valuable resource for both novice and experienced readers interested in CF-based RS.

Figure 1 showcases a comprehensive overview of the most significant traditional methods, techniques, and algorithms utilized in the recommendation process, along with their interrelationships and groupings. This figure offers a compelling glimpse into the complex world of recommendation systems, which have become increasingly vital in today's digital landscape. This paper provides in-depth insights into the various critical aspects involved in the recommendation process, allowing readers to develop a deeper understanding of how these complex systems operate. As illustrated in Figure 1, traditional filtering methods such as content-based and CF can be applied to databases, with model-based technologies like MF, Genetics, clustering, Association rule algorithms, and neural networks leveraging this information to enhance the accuracy of recommendations. Memory-based approaches, including item-to-item, user-to-user, and hybrid models combining both, represent another popular recommendation technique. The ultimate goal of both memory-based and model-based approaches is to deliver the most precise predictions of user preferences. Evaluation of recommendation system accuracy is often carried out using classical information retrieval metrics such as MAE, precision, and recall. Researchers use these measures to continually refine CF methods and technologies, ensuring that these systems continue to evolve and improve. Overall, the findings presented in this paper offer significant insights into the complex world of recommendation systems, providing a vital resource for researchers, practitioners, and anyone seeking to harness the power of these systems to enhance their digital experiences.

4. CF algorithm foundations

Collaborative filtering (CF) is the method through which information is recommended to users based on the wishes of a group with similar interests and experiences. This procedure is separated into two phases in CF algorithms: the online and the offline. The online process is to identify products that may be liked by users online, but in the offline phase, certain data that is not recommendable, such as low-recommendation data or information the user acquires, while the recommendation value is high, will be filtered away. The CF algorithm has been widely utilized since it does not specifically need suggestions and can handle unstructured and complicated objects, such as music and films. The CF creates and automates matching suggestions for users so that the user's recommendation is implicitly obtained from the buying or browsing mode of the system, and the user does not have to work hard to discover the right interests. Information is recommended, such as completing certain survey forms. CF algorithm may find prospective interests and preferences of users that have not yet been found. These benefits make it appropriate for virtually all areas.

4.1. Collection of information in CF algorithms

To be relevant, a RS must be able to make predictions about user interests. It is therefore necessary to be able to collect a certain amount of data on these to be able to build a profile for each user. A distinction can be made between two forms of data collection: explicit data collection and Implicit data collection.

Explicit data collection is also called active filtering, which relies on the user explicitly indicating their interests to the system. For example, asking a user to comment, tag, rate, like, or even add as favorites content (objects, articles, etc.) that interests him [45]. Often used is a scale of ratings ranging from 1 star (I do not like at all) to 5 stars (I love), which are then converted into digital values to be used by the recommendation algorithms (see Figure 2 A). The Advantage of explicit data is the ability to reconstruct an individual's history and the ability to avoid aggregating information that does not correspond to this single user (several people on the same workstation). The disadvantage is that the information collected may contain a so-called declaration bias.

Implicit data collection is called also passive filtering, which is based on an observation and an analysis of user behavior carried out implicitly in the

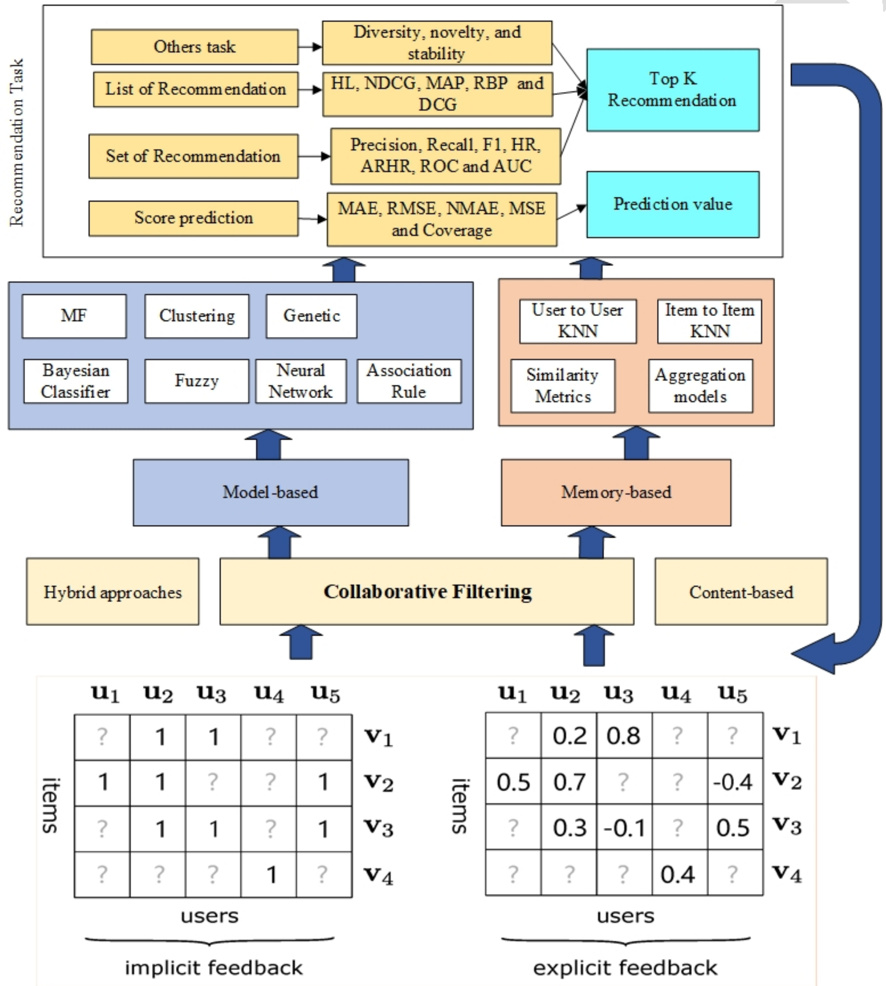


Figure 1: Collaborative filtering Model Overview.

application that embeds the RS, all done in the "background" (basically without asking the user anything). For example, get a list of items the user has listened to, watched, or purchased online. Analyze the frequency of consultation by a user and the time spent on a page. Monitor the user's online behavior. Analyze your social network [45]. The Advantage of implicit data is that no information is requested from users, all information is collected automatically. The data retrieved are a priori correct and do not contain any reporting bias (see Figure 2 B). The disadvantage is that the data retrieved is more difficult to attribute to a user and may therefore contain attribution bias (common use of the same account by several users). A user may not like some books they bought, or they may have bought them for someone else. In CF RS, two forms of CF methods, namely memory-based and model-based methods, are used to process explicit and implicit data. These algorithms are listed in the following sections.

Figure 2: A Simple example showing the difference between explicit and implicit feedback rating matrices [45].

4.2. Memory-based CF

Memory-based CF finds similarities between users or items based on the user-item rating matrix and recommends them to the active user [22]. This type of CF algorithm is called a neighborhood-based algorithm. No model training is required, and recommendation results can be generated for users based on a very simple idea. There are two main types: user-based CF (UCF) and Item-based CF (ICF). A UCF algorithm is adjacent to obtaining similar hobbies or interests with similar statistical methods for a user. The ICF method has a basic assumption that "items that can arouse user interest must be similar to the previous high-scoring items", and replaces the similarity between users by calculating the similarity between items [46, 47, 48, 49].

Their common shortcoming is that the data is sparse and difficult to process. Large amounts of data affect immediate results, so model-based CF technology has been developed. Model-based CF analyzes historical data first to create a model and then makes predictions based on this model. CF Model-based technologies are extensively utilized, such as matrix factoring, Bayesian networks, etc. Latent semantic indexing. Based on sample analysis, the model is obtained.

4.3. Model-based CF

The model-based CF is referred to as a CF learning machine. CF builds a parameter model to characterize the user or item connection and then optimizes the model parameters. It is scalable and more sparse than the memory-based approach. The mainstream methods of model-based can be divided into using association algorithms, clustering algorithms[50], classification algorithms [51], regression algorithms [47], matrix factorization [52], neural networks [53], graph models, and implicit semantic models to solve them. We introduce them separately below.

4.3.1. Matrix factorization for CF

The most common CF approach is matrix factorization (MF). MF algorithms convert users and items into vectors of the same dimension that represent the latent features of each. In general, MF approaches are more effective because they allow us to discover the latent features that underlie the user-item interaction. There are two or more factors that can be found that multiply together to form the original matrix, which is the goal of factorization [52]. Training such a model can be effectively solved by using SGD [52] or alternating least squares (ALS) [54] to minimize the sum squared distance. A very large number of research papers have been published on this subject, addressing various extensions and variations. This area is still very active among ML researchers. Multiple extensions and variations of this topic have been the subject of numerous research papers. ML researchers are still very active in this area. Innumerable proposals have used MF for CF, but they differ in their definitions of the loss function and regularization function.

In the literature, several loss functions have been proposed, including the square-loss function [55], Bregman divergence [56], logistic log-likelihood loss [57], KL-divergence [58] and hinge-loss(smoothed hinge loss) [59]. Overfitting must be avoided when the matrix completion problem is viewed as supervised learning. We used the regularisation term to penalize overfitting caused by large latent factor values.

4.3.2. Clustering for CF

CF candidates can be reduced using clustering techniques. Using a clustering algorithm for CF is somewhat similar to the previous CF based on users or items. We can cluster according to users or according to items based on a certain distance metric. If based on user clustering, and cluster users with a certain distance measurement method. Here we can use the similarity measurement method, and then recommend the products of users in the same cluster to other users. The items of these users are gathered, and then TOP n is taken as a user recommendation. If a user has never purchased an item, but the items purchased by other users in the same cluster will be recommended to the user, it is suitable for recommendation by new users [60]. Item-based clustering is similar to user-based clustering, except that users are recommended for products. All users in the product cluster share these products. Although some products have never been purchased, this method can help find the target group recommended for new items[61, 62]. Commonly used clustering recommendation algorithms include K-Means, Self-Organizing Map (SOM), BIRCH, DBSCAN, and spectral clustering.

K-means and Self-Organizing Map (SOM) algorithms are the most commonly used clustering methods [63, 64, 65], according to authors in [66], K-means takes an input parameter and partitions n items into K clusters. A method for unsupervised learning based on artificial neuron clustering, the Self-Organizing Map (SOM), was developed by the authors in [67]. The disadvantage of the clustering algorithm is that it is full of new content and new users, and these users and content will never have recommendations. For example, a new user is a cluster group, and these new users cannot recommend them if they have not purchased products. Products may be purchased in the later purchase process, as long as the user has some behavior, the user can be recommended later.

4.3.3. Association rule for CF

Association rule mining is usually a general technology that appears in similar rules or relational patterns in large-scale transactions. This technique can also be applied to the RS of the CF algorithm. Association rules are also concepts in data mining. It analyzes data and finds the association between data, that is, the correlation between various data [68, 69]. The most commonly used algorithm in this type of algorithm is the Apriori algorithm [70], and the measurement indicators of association rules are support and confidence. Using this principle can avoid the exponential growth of the number of item sets, thereby calculating frequent item sets in a reasonable time, saving more processing time for a recommendation.

More precisely, the main problem encountered in applying association rules to the RS is converting scores into transactions 0 or 1. Usually, all positive score sets (that is, the score matrix after averaging in the past) or the user's preferences, such as purchase behavior or record, are regarded as transactions. When using the Apriori algorithm, which reduces the size of frequent sets and provides good performance by utilizing Apriori properties, candidate itemsets are generated. These methods and goals do, however, have a clear connection, but they have not yet gained widespread acceptance. Each iteration has to traverse the entire data for counting calculations and judgments, and the operating efficiency is low.

4.3.4. Classification for CF

With user ratings as the basis for segmentation, this problem is transformed into one of classification. Setting a score threshold, for example, is the simplest method. It is recommended if the score is higher than the threshold, and it is not recommended if the score is lower than the threshold. We convert the problem into a two-category problem. Although there are so many algorithms for classification problems, logistic regression is currently the most widely used. Why is it a logistic regression instead of a taller-looking support vector machine? Because logistic regression is more explanatory, we have a clear probability of determining whether each item is recommended or not. At the same time, we can engineer the characteristics of the data to achieve the purpose of tuning. Common classification recommendation algorithms include logistic regression and naive Bayes, both of which are characterized by strong interpretability. The advantage of the algorithm is that it is simple, fast, and Explainable. While the disadvantage of the algorithm seems to be that user characteristics are not considered, and commodity-related characteristics are completely constructed. Of course, user characteristics can also be considered commodity characteristics.

4.3.5. Others algorithms for CF based on ML

Among the most widely used models, we have fuzzy systems, genetic algorithms [71], and implicit semantic model, among others. One of the most widely used fuzzy clustering methods is the fuzzy C-means (FCM) algorithm [72]. Contrary to conventional K-means clustering, FCM clustering establishes a fuzzy boundary around each cluster to which objects will be bound in the form of memberships, $u \in [0, 1]$. The clustering problem is solved by optimizing JFCM, an objective function that measures the distance between points and cluster centers [72]. According to the authors in [73], the GAs have been utilized in three aspects: clustering, hybrid user models, and using the GAs without requiring the additional information provided by the hybrid model [73]. Usually, the GA is used to find the optimal similarity metric [71]. For scoring recommendations, NLP is used as the primary basis for the implicit semantic model [43]. The main methods include implicit semantic analysis LSA and implicit Dirichlet distribution (LDA).

4.4. DL for CF

The past few years have witnessed a tremendous surge in the volume of research devoted to DL-based CF RS. This section offers a comprehensive exploration of the underlying rationales and methodologies behind using DL techniques in CF. We delve into several approaches in this section, including Boltzmann Restricted Machines (RBM), Deep Belief Networks (DBN), Autoencoder, Recurrent Neural Networks (RNN), and Convolutional Neural Networks (CNN). Additionally, we examine some lesser-known but intriguing methods in the subsection on Other approaches.

4.4.1. RBM for CF

The Restricted Boltzmann Machine (RBM) is a widely used generative neural network model in the field of unsupervised learning. Its primary objective is to learn the joint probability distribution of both visible and hidden units, with the hidden units representing latent features and the visible units linked to input data. Once the model has learned the joint distribution, it can generate new examples using sampling. In [74, 75] introduced an RBM model with only two layers and presented the learning and inference process using user ratings for movies. As a result, RBM can be successfully applied to the Netflix dataset. Users' evaluation of items or the relationship between users can be modeled to improve the accuracy of the RS, for instance, in [76], the authors used RBMs to jointly learn both the relationship between such a user's ratings of items and the relationship between such users that scored a certain item. In [77], the authors used RBMs approaches to modeling group preferences in group-based RS by jointly modeling group characteristics and group profiles. RBM approach has also been used to project the pairwise association between items or users, for instance, through the connections between layers and the softmax layer, in [78], the authors determined the relationship between a rated item and its rating as well as relationships among rated items. Consequently, the RBM approach was used for combining side information, for instance, in [79], the authors linked the machine parameters with side information. Mainly, RBM approaches are utilized to support a low-rank representation of user preferences. Furthermore, RBM approaches are used for combining UCF or ICF integration, as well as neighborhood formation within the visible layer. For instance, using RBMs, in [76], the authors proposed a hybrid approach of UCF and ICF based on RBM. It is still more accurate to use Boltzmann machines to project user or item correlations and neighborhood formation. RBMs may be preferable to Boltzmann machines when considering pairwise user and item correlations because they have simpler parametrization and are more scalable [76], for instance, in [80], the authors used RBM to treat CF limitations and proposed the neighborhood-conditional RBM (AND RBM) model, which is based on joint distributions of similarity and popularity scores. In [81], the authors proposed an RBM method extension for CF with implicit feedback. In [82], the authors suggested a CF model Using RBM and Fuzzy C-means (FCM). The proposed model uses the FCM clustering algorithm to group users, and then RBM is utilized to predict the users' preferences. The method proposed by the authors in [83], takes item preference relations as input and generates a ranking of items for any user. Conditional RBM incorporates item-side information as well as preference relations into the method. The RBM method is used to represent entities for pairwise collaborative ranking [84]. In [85], the authors examined the effectiveness of conditional RBM (CRBM) in modeling users' rating preferences for the top-k recommendations from a new perspective. In [86], the authors used visual features in conjunction with RBM to address the New Item problem in video RS. In [87], the authors propose a Web-based movie RS that uses various recommendation algorithms to recommend movies to users based on their profiles. To optimize the RBM parameters for CF, in [88], the authors proposed a hybrid algorithm that combines cluster-based RBM and differential evolution (DE). RBMs can also handle large datasets.

4.4.2. DBNs for CF

DBNs are utilized in CF along with content-based filtering to extract hidden and valuable audio feature characteristics for music recommendation [89]. DBNs are also used in CF RS that use text [90, 91]. It is also used in content-based RS to evaluate user preferences, especially for textual data [90]. DBNs are used to provide a semantic representation of words [91]). Furthermore, based on user preferences, DBNs are used to extract high-level features from low-level features [77]. These studies show that DBN algorithms are mostly used for feature extraction and classification tasks, particularly on music and text data.

4.4.3. Auto-encoders for CF

The fundamental architecture of an auto-encoder consists of two key processes: encoding and decoding. During encoding, the input is transformed into a vector within a low-dimensional hidden space, which is then decoded by the decoder to reconstruct the original input. The neural network structure reveals that the encoding involves a linear combination followed by a non-linear activation function. Without this nonlinear transformation, the auto-encoder would be equivalent to standard PCA. Auto-encoder techniques play an essential role in recommender systems, as they enable the training and retraining of a nonlinear function to handle missing values in the user-item rating matrix[92, 93]. In addition to dimensionality reduction, the auto-encoder approach is also used to extract more latent features by utilizing the encoder parts' output values [75, 94, 95]. Moreover, Sparse coding is also used by the auto-encoder to learn more effective features [94]. For instance, auto-encoders with a DAEs model were proposed by [96]. Because the input data is corrupted, the identity network cannot be created. DAEs are used to predict missing values from corrupted data in RS. A further benefit of SDAEs is that they assist RS in determining a denser form of the input matrix, as well in [97]. DAE models are simply multiple auto-encoders stacked on top of each other. Moreover, in [98], the authors proposed stacked demonizing auto-encoders (SDAE) and probabilistic matrix factorization (PMF) models. These auto-encoders make it possible to extract additional hidden features. In [99], the authors proposed a depth MF. The conventional MF method is utilized to decompose the user and item feature matrices. A multilayer feed-forward neural network is used to mine associated features deeply. The RS's predicted rating is the inner product of the corresponding low-dimensional features.

By combining the video data features learned by shrinking the auto-encoder with the implicit feedback gathered by SVD++, in [100] the authors proposed an Auto SVD++ model that would improve recommendation accuracy. In [92], the authors proposed CF DL based on auto-encoders (ACF). It divides the rating value of an item based on user ratings into five vectors. Integer scoring prediction is a problem that this method does not address, which increases the sparsity of the scoring matrix and reduces prediction accuracy for the ACF algorithm. In addition, in [93], the authors developed the AutoRec model which reconstruction of input data is the model's primary objective. As a result of its ability to handle non-integer prediction scoring values, the AutoRec model is less susceptible to overfitting because it does not introduce noise into its input. To predict ranking, in [96], the authors proposed the CDAE model. The implicit feedback data from the user on the items is the input to this model. Perceptrons in the input model correspond to specific items, and the user's preference for an item is indicated by the values 0 or 1. Subsequently, the items associated with the predicted values of the model's output layer perceptron are recommended to the user in sequential order. According to the authors in [101], one of the restrictions of the preceding models is a cold start issue. Besides that, by allowing data from multiple data sources, the auto-encoder aids in the interoperability of side information into a CF RS[102, 98, 103, 104, 105, 106]. For instance, in [98] the authors used Bayesian SDAE and [103] the authors used marginalized DAE to incorporate side information into their RS. In [102] the authors proposed relational SDAE, which entails producing a probabilistic form of SDAEs to incorporate side information with ratings.

Side information is only integrated at the input level in [98]. In contrast to the authors in [98], side information is integrated into each layer of SDAEs in [104]. To resolve the sparsity problem, the authors retrieved features from side information using PMF and time SVD++ auto coders by the authors in [104] and [106]. In [104], the authors developed a CFN method that includes extra information and a scoring matrix to generate final prediction results. In [106], the authors employed SDAEs to extract latent hidden features of extra information and incorporate them into the Bayesian model of the pairwise ranking model. The model's recommendation accuracy has improved when compared to previous methods. According to the authors in [101], the limitations of this model are that the side information is comparatively simple and the data is very sparse.

By switching the SVD algorithm to a Stacked Denoising Auto-encoder, the authors in [107] proposed CF to supervised learning (COFILS) to reduce data sparsity and allow testing of a variety of supervised learning algorithms rather than matrix factorization approaches (SDA). In [108], the authors proposed a Bayesian generative model called collaborative variational auto-encoder (CVAE) for a multimedia recommendation that takes into account both rating and side information. In [109] the authors extended VAEs to CF with side information, such as when ratings are margined with explicit user text feedback. Variational auto-encoders for CF have been proposed. They introduced a new regularisation parameter for the learning objective, which is critical for achieving competitive performance [110]. In [111], the authors developed a Collective Variational Auto-encoder for Top-N Recommendation with Side information. In [112], the authors proposed an additional variational auto-encoder that takes into account both the item's side information and its tag feedback. Unsupervised, the model learns effective latent representations from additional side information. For the top-N RS task, in [113], the authors proposed an improved collaborative auto-encoder with knowledge distillation. By combining simple elements from the literature, in [114], the authors defined a linear model geared toward sparse data, specifically implicit feedback data for RS. In [115], the authors proposed a multi-model DL (MMDL) approach for constructing a hybrid RS with significant improvement by integrating user and item functions.

In [116], the authors suggested technique incorporates CF and content filtering, additional user social impact. Each user's social influence is determined based on his or her social traits and Twitter behaviors. In [117], the authors proposed a fast deep auto-encoder for high-dimensional and sparse matrices. In [118], the authors used to generate more accurate recommendations by exploiting the non-trivial, nonlinear, and hidden relationships between users in terms of multi-criteria preferences.

According to research in this field, auto-encoders provide more accurate recommendations than RBMs. One reason for this is that RBMs predict by maximizing log-likelihood, whereas auto-encoders predict by minimizing RMSE. For auto-encoders, gradient-based back-propagation is used, whereas contrastive divergence is used for RBMs. Because stacking auto-encoders allows for deeper learning of more hidden features [93], they provide more accurate predictions than non-stacked forms. Feature extraction, dimension reduction, and prediction are just a few of the uses for auto-encoders in RS

[103]. In RS, auto-encoders are used to deal with sparsity and scalability issues.

4.4.4. RNNs for CF

RNNs are taught to process data in a specific order. In an e-commerce system, a user's current browsing data influences her purchase behavior. Most typical RS, on the other hand, produces user preferences at the start of a session, ignoring the current history and the order of sequences of user actions. In RS, RNNs are used to incorporate current viewing homepage records and requests of views to support further accuracy of prediction [119, 120, 121]. For instance, in [119], the authors combined RNN results with the feed-forward artificial neural results obtained to generate predictions that took user-item correlations into account, in [122] the authors combined RNNs with latent user preferences to represent temporal and contextual aspects of user behaviors to enhance recommendation accuracy. RNN approaches are also used to demonstrate the influence of users on latent features of items and their co-evolution over time in a nonlinear way [123]. By viewing the problem as a sequence prediction problem, in [124], the authors used RNNs to incorporate the evolution of user preferences into the recommendation process. Several conclusions can be drawn from studies on DL on RS. In terms of recommendation coverage and short-term predictions, RNNs outperform standard neighbor-based and MF methods (predicting the next consumable item).

This success can be attributed to RNNs that account for user taste evolution and also the co-evolution of user and item latent features [124, 123]. Aside from that, RNNs are ideal for session-based RS and combining users' implicit actions with their preferences. Another RNN-based idea by the authors in [125], the exploration of various strategies for integrating user long-term preferences with session patterns encoded by recurrent neural networks (RNNs). In [126], the authors proposed a recommendations model called slanderous user detection RS (SDRs) to address the slanderous user detection problem as a multi-view unsupervised problem and improve the recommender system's performance. In [127], the authors employed RNNs for both long-term and short-term sequential recommendations. As a result, RNNs are suitable for session-based RS and integrating users' implicit behavior patterns with individual preferences (like or dislike).

4.4.5. CNNs for CF

The most common type of neural network used in image processing and computer vision is the convolutional neural network (CNN). CNNs can also help with RS. CNNs have fewer parameters than MLPs with the same number of perceptrons, making them easier to train. CNNs are made up of convolutional layers, a pooling layer, and fully connected layers. [128]. Convolutional layers extract features from the input and produce n feature maps, where n is the number of filters. The pooling layer is responsible for reducing the dimensionality of features to address the issues associated with feature maps that have a high degree of dimensionality. In [129], the authors presented the extraction of latent factors from audio using CNN method-based information feedback. In [130], the authors employed CNN methods to identify latent factors in text data. In [131], the authors extracted visual features to generate visual interest profiles of users for recommendation. In [132] the authors used CNNs methods to extract latent features from images so that the features and user behaviors could be mapped into the same latent space.

The ConvENT model combines CNN with probabilistic matrix factorization to use document contextual information to address data sparsity issues and improve prediction accuracy [133]. CNNs are only intended to work with two-dimensional data, such as images and videos. As a result, they are frequently portrayed as 2D-CNN. Recently, researchers have been drawn to 1D Convolutional Neural Networks (1D-CNNs), a modified version of the 2D CNN model (1D-CNN) [134]. Many studies and applications have shown that 1D-CNN models are more powerful than their 2D counterparts when dealing with 1D features. This is because forward and backpropagation in 1D-CNN necessitate simple array operations [135]. This means that 1D-CNNs have much lower computational complexity than 2D-CNNs. According to recent research, shallow architecture 1D-CNNs (i.e., a small number of hidden layers and perceptrons) can learn difficult tasks involving 1D features [135]. In contrast, the 2D-CNN model necessitates a more profound, in-depth architecture for training and implementation. 1D-CNN is well-suited for real-time and low-cost applications such as mobile and handheld devices due to its lower computational requirements.

4.5. Limitations of CF

In comparison to CBF, CF has several major advantages (such as opinions and ideals). If the user's profile doesn't contain relevant content, the CF technique can provide them with serendipitous suggestions [136]. Given the advantages of CF techniques, their widespread application has revealed some potential issues, such as those listed below.

4.5.1. Cold-start problem

The first-rater or new-item problem is another name for the cold start problem. In some ways, it might be viewed as an extreme case of the sparse problem. Because traditional CF-based recommendation depends on calculating similar users/things to acquire the target user's suggestion, when a new item emerges for the first time, the simple CF-based RS cannot forecast its score and recommend items because no user has evaluated it. Furthermore, because new things are introduced so quickly, there are fewer user reviews, and the accuracy of recommendations is likewise relatively low. Similarly, the mechanism for recommending new users is deplorable. When a CF-based RS runs for the first time, every user encounters the cold start problem on every project, which is an extreme case of the cold start problem.[36, 25, 137].

4.5.2. Data sparsity problem

To represent user information in the implementation of CF technology, a user-item evaluation matrix must first be used. However, this sounds simple in theory, several more e-commerce RSs require a huge amount of data information that can be processed, but in these systems, the total amount of goods purchased by general users accounts for about 1% of the total amount of goods on the website, so the evaluation matrix (user-item matrix) is very sparse. On the one hand, it is difficult to find the nearest neighbor user set in the case of large and sparse data, and on the other hand, the cost of similarity calculation is very high. At about the same time, because the data is sparse when the target user's nearest neighbor user set is formed, moving forward will result in information loss, reducing the recommendation effect. For instance, consider the transitive loss of neighbor-user relationships. User A has a high correlation with user B, and user B has a high correlation with user C. However, because users A and C rarely evaluate the same product, they believe the degree of correlation between the two is low due to the scarcity of data. The potential relationship between users A and C has been shattered [32, 36, 37].

4.5.3. Scalability problem

CF-based RS platforms frequently have millions of users, products, and content. As a result, offering suggestions to users necessitates a significant amount of computing power. The user-to-user type of social recommendation is also known as memory-based because the rating database is kept in memory on the server at all points in time and is used directly to generate recommendations to the active user. However, while this memory-based approach is theoretically more precise, it suffers from scalability issues because it has all of the data available permanently and in real-time to generate the recommendations. For databases with millions of users and millions of pieces of content. The data is first processed offline in an item-by-item or model-based approach. The "learned" or preprocessed model will then be used to make predictions when the web application or service is run. The model-based approach avoids the scalability issue [137, 138].

4.6. Evaluation Metrics (Recommendation Tasks)

The assessment of prediction and recommendation outcomes has always been a critical connection since the beginning of RS research. The advantages and downsides of RS are directly reflected in its performance on these assessment metrics. The most widely used recommendation quality measuring methods may be classified into four groups based on the distinct recommended tasks:

1. **Score prediction Metrics**: Standardized Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Coverage, and Normalized Mean Absolute Error (NMAE) are among the most representative metrics for the score prediction task.
2. **Metrics of the set of recommendations**: Examine the predicted item set to see if it is eligible for Top-N recommendation tasks. It might be difficult to estimate the user's rating of an item directly due to data sparsity and cold-start issues. As a result, some researchers have proposed a Top-N recommendation technique that is based on the user-item rating rather than the user's assessment of the item. To produce a collection of items that the user is most likely to like and recommend to the user, implicit interaction (such as clicking or favoring) is used. Precision, Recall, F1, Hit Rate (HR), ARHR, ROC, and AUC are a few examples.
3. **Metrics of the list of recommendations**: The recommendation effect is weighted and evaluated according to the ranking list, which can be applied to both the score prediction task and the Top-N recommendation task. When the number of recommended items is large, users

will pay more attention to the items ranked first in the recommended list. The errors that occur in these items are more serious than the errors in the items next to them. The method of weighted evaluation of the recommendation effect according to the ranking list considers this situation. Among the most commonly used ranking metrics, such as Half-life (HL), Normalized Discounted Cumulative Gain (NDCG), Mean Average Precision (MAP), Rank-biased precision (RBP), and Discounted cumulative gain (DCG)

4. **Other evaluation Metrics**: Most articles discuss methods that try to improve the accuracy of recommendation results (RMSE, MAE, etc.) for scoring prediction tasks. It is also common to try to improve the accuracy, recall, and ROC of Top-N recommendations. However, to obtain higher user satisfaction, other goals should also be considered, such as subject diversity, novelty, and stability of recommendations.

5. Remedies for challenges of CF-based-based RS

In this section, we conduct a thorough evaluation of studies focused on addressing issues related to CF model-based The first subsection outlines remedies for challenges based on machine learning-based CF-based RS algorithms, while the subsequent subsection delves into remedies for challenges based on DL CF-based R algorithms.

5.1. Remedies for challenges of CF-based on ML Models

One way of using ML in CF models is to produce effective solutions to CF-based RS problems. In this subsection, we examine ML models that offer guidance to the problems of RS.

5.1.1. Remedies for Data Sparsity and Scalability of CF-based RS Algorithms

The user-item rating matrix is handled by the majority of existing clustering-based recommendation methods to divide users or items into groups. That is, they compute rating-based similarity, i.e., user-user or item-item similarity, and then use a clustering algorithm to generate user or item groups to improve recommendation quality due to data sparsity and high dimension [139, 140, 141, 142, 143]. These models, however, require additional user or item properties to group users into clusters. In most cases, obtaining these characteristics in a practical application is difficult. The system can make inferences based on the underlying reasons why a user may or may not be interested in a given item using integrated semantic similarities for things with ratings or user-based similarities [69]. In [144], the author improved the traditional similarity measures by employing the information entropy of user ratings so that the user's global rating behavior on items can be reflected. It combines entropy with traditional similarity measures. As a result, the impact of a rating difference between two users on similarity varies depending on the rating records on the item. This process reduces the poor recommendation quality caused by data sparsity and high dimensions. In [145], the author demonstrated an intelligent RS that is based on a user clustering CF approach with grey wolf optimization. In [146], the author developed a CF algorithm that is based on user preference clustering. To address the limitations of traditional similarity, they proposed a new similarity measure method that takes into account user preference from both local and global perspectives. In [147], the author improved prediction accuracy and effectively dealt with the sparsity problem by proposing the K-medoids clustering RS for the CF algorithm based on probability distribution. In [148], the authors proposed a hybrid model based on a movie RS that employs K-Means, the particle swarm optimization (PSO) method, and fuzzy c-means (FCM) to classify movie types based on users, resulting in reduced computation complexity on a large dataset. In [149], the authors used local similarity based on affinity propagation and K-means, which improves prediction accuracy in neighborhood-based CF. In [150], the authors switch hybrid methods for predicting user ratings that use either our custom similarity measure or user and item biases. In [151], the author proposed a normalization-based CF recommender to overcome the sparsity problem.

In [152], the author demonstrated a new bio-inspired clustering ensemble for UCF by combining swarm intelligence and fuzzy clustering models. In [72], the author presented a method for detecting noisy ratings based on fuzzy clustering. To represent the data-driven uncertainty, the entropy of a subset of the original rating dataset is used, and evaluation metrics are used to represent the prediction-driven uncertainty. In [60], the author proposed a clustering-based CF model in RS that makes careful use of an incentivized/penalized user model (IPU).

Models of biclustering that combine the UCF and ICF models to project dimension reduction [153, 154, 155]. The similarity between users or items is computed using biclustering based on the cluster to which they belong. The biclustering method samples data and creates biclusters of users and items based on similarity in rating patterns. Another hybrid model combines the

UCF and ICF models to improve scalability and sparse data [47, 156, 157, 158, 159, 160, 161, 162, 163].

For instance, In [156], the author proposed a hybrid model based on a Bayesian Classifier that combines the UCF and ICF [157] proposed a hybrid model that combined UCF and ICF prediction results with a smoothing sparse data (HSPA) model. In [161], the author proposed a hybrid model that combines the prediction results of the possibilities UCF and ICT model using the possibilistic information affinity measure to represent user-item preferences. [164] proposed a multi-factor similarity metric that captures the linear and nonlinear correlations between users caused by extreme behavior. Following that, a fusion method in a probability MF framework is proposed that takes into account both multifactor similarity and global rating information. To improve the prediction accuracy of CF methods, for instance, in [165], the author proposed a hybrid similarity to be used in a KNN memory-based CF model. This is accomplished without sacrificing the benefits of memory-based CF methods, which require less memory and time than model-based CF methods. In [166], the author enhanced the CF-based RS using dimensionality reduction and clustering approaches. The suggested model employed the k-means method and SVD to cluster related users and minimize dimensionality.

To deal with the dimension reduction some commonly used matrix MF models include Non-negative MF (NMF) [167], SVD [34], incremental SVD based MF [168], weighted low-rank approximation [169], Non-parametric probabilistic principal component analysis (NPCA) [170], matrix completion technique in the Netflix problem as a practical example for the utilization of the technique [171], and SVD technique used in an OptSpace algorithm by [172]. In [173], the author developed the probabilistic MF (PMF), which scales linearly on large data sets and outperforms traditional SVD models. Several variants and generalizations based on PMF are proposed, such as Bayesian PMF. [174], and [175] incremental CF recommender based on regularized MF. A multifaceted CF Model (SVD++) by [176], Pairwise probabilistic CF for implicit feedback CF (PPMF) by [177]. In [178], the author proposed two new initialization models for PMF which take into consideration the distribution statistics of user product ratings to enrich latent vectors. To deal with multi-level ordinal rating data, Maximum Margin MF (MMMF) by [179], ensembles of maximum margin MF (E-MMMF) by [180], Nonparametric max-margin MF for collaborative prediction (iPMMMF and iBPMMMF) by [181], fast max-margin MF with data augmentation (Gibbs

MMMF and iPMMMF) by [182], and multiple bi-level MF (MMMFs) [183]. However, MF model methods produce accurate predictions but are computationally expensive; they can only be used in static offline settings where the known preference information does not change over time [184].

Moreover, the MF technique's main problem is that it identifies only linear interactions between user and item latent features, which may not be adequate to model the actual interaction between them. The well-known cold start problem for new users and items is another issue with MF algorithms [185]. In [186], the author proposed an effective Adaptive-Support Association Rule Mining (ASARM) system for RS. This algorithm took advantage of the fact that the mining task is specific by tying the rule's head to the target user or item. In [187], the author proposed a model that used multi-level association rules to alleviate the data sparsity and scalability of CF. In [188], the author proposed a CF model based on Fuzzy Association Rules and Multiple-level Similarity (FARAMS). FARAMS extended existing techniques by using fuzzy association rule mining. In [189], the author proposed enhancing CF recommendations by utilizing Multi-Objective Particle Swarm Optimization (MOPSO) embedded association rule mining. [190]proposed a genetic algorithm-based efficient method for producing cred association rules with higher performances. The genetic algorithm is used to find association rules when there is no minimum support. When the search space is too large, the genetic algorithm is efficient. In [191], the author used clustering and association rules mining on implicit data to improve the accuracy of CF recommendations. In [192], the author presented the maximum posterior estimator derived from a Bayesian analysis of the problem as a generic optimization criterion, BPR-Opt for personalized ranking. In [193], the authors presented two probabilistic models with prior parameters that the user can set to encourage the model to have a desired size and shape, to conform to a domain-specific definition of interpretability. In [194], the author proposed a probabilistic model that incorporates items, users, and the relationships between them into a generative process. They create an ensemble of collaborative filters using a progressive algorithm. In [195], the authors proposed a distributed algorithm for large-scale Bayesian PMF that is based on a newly discovered principle of distributed variational Bayesian inference. CF model called CFNBC for inferring potential lncRNA-disease associations is proposed based on the Naïve Bayesian Classifier on two datasets of known miRNAdisease associations and miRNA-lncRNA associations by [196]. In [197], the authors proposed the BNMF technique for predicting user preferences in RS based on CF. It works by factoring the rating matrix into two non-negative matrices with understandable probabilistic meanings.

To improve the accuracy of recommendations, in [65], the author proposed two hybrid methods that combine the benefits of the WRFM-based method and the preference-based CF method. In [64], the author tested a variety of partitioning strategies in tandem with a dimension reduction strategy. In [198], the author proposed a LeaderRank-based initialization method for the k-means clustering algorithm with side information. In [66], the author proposed a hybrid model that partitioned transformed user space on the MovieLens dataset using K-means clustering and genetic algorithms (GAs). Personalized suggestion CF-based system for IoT scenarios by [199]. They proposed the TCCF recommendation model, which is based on the time correlation coefficient and an improved K-means with cuckoo search (CSK-means) for faster and more accurate recommendations.

5.1.2. Solutions for cold start problems of CF based on ML Models

Incorporating side information is an intuitive solution for the cold start problem. In most cases, the users and items have available demographic information that can be used as relevant side information. In the case where a user has no interactions (examples include new users or existing users trying out a new category of products) but still some available side information like age, gender, occupation, etc. In several studies that explore the MF to develop a high-quality user-item rating matrix, researchers are examining various MF approaches with side information linked to users' implicit or explicit feedback. For recommendations that include both explicit and implicit input, for instance, [200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211], the authors take into account three explicit feedbacks such as ratings, helpfulness score, and centrality score, as well as one implicit feedback such as the view relationship with the user-item. In [207], the authors utilized side information by combining an arbitrary nonlinear quantile regression model and MF under a Bayesian framework, explicit feedback meets implicit feedback in generalized probabilistic MF mode (GPMF) by [206], in [201], the authors combined content-based filtering with CF, harnessing the information of both ratings and reviews. In the context of travel-product recommendation, in [208] the authors combined the PMF with the multi-auxiliary information (PMF-MAI) model. In [209] the authors proposed a PMF model that uses preference relations as input (rather than ratings) to generate an efficient item ranking. The matrix co-factorization method is used to incorporate user and item-side information into the model. In [210], the authors propose a probabilistic framework for combining mixed-data features like MF with multimodal side information. In [211], the authors took three social factors into account: personal interest, interpersonal interest similarity, and interpersonal influence, which they combined into a unified personalized RS based on probabilistic MF. For cold start recommendations, the authors in [212], proposed the aspect model latent variable method. In model fitting, the aspect model uses a Bayesian classifier to combine collaborative and content information. Cross-Level Association RulEs (CLARE) were used by [213] to integrate content information about domain items into collaborative filters. To address the cold-start problem in RS, they introduce a preference model that includes both user–item and item–item relationships. In [214], the authors developed an associated foods recommender algorithm to remind respondents of omitted foods and improve the ranking quality of search results returned in response to respondents' free-text food name queries. In [215], the authors proposed a heuristic similarity measure aimed at improving recommendation performance under cold-start conditions in which only a small number of ratings are available for similarity calculation for each user. In [216], the authors proposed a new similarity measure that was perfected through optimization based on neural learning to function properly in cold-start situations. In [217], the author's Pre-computed models, namely the error-reflected model, are built by reflecting the prediction errors. To overcome the cold-start problems [218] proposed a model that combines content metadata with traditional user-content ratings and trust network data. In[219], the authors proposed using the temporal (ratings per year basis) variance of Top-N Neighbors to improve the accuracy of CF-based recommendations. On years, optimized k-means clustering is used to find a cluster of similar users in an optimized cluster of years. In [220], The authors employed neighborhood reduction to overcome the ICF model's sparsity and cold start problems.

Some works on clustering handle cold start problems, for instance, in [221], the authors optimized the standard MF by incorporating the user clustering regularisation term. This model considers both user-item rating information and user information. There are also works that focus on the simultaneous clustering of users and items (bi-clustering) to improve prediction quality and reduce the cold-start problem when applied to hybrid filtering [222, 223, 224, 225, 226, 227, 228]. In the same vein, some researchers are investigating the clustering of social trust information to improve RS. To alleviate the cold-start problem, hybrid approaches such as clustering and filtering are being considered. For instance, in [222], the authors proposed a clustering model for CF recommendation based on social network analysis. In [226], the authors proposed a probabilistic method for deriving a prediction from the perspectives of both ratings and trust relationships. In [229], the authors proposed an explicit trust and distrust clustering-based CF recommendation method. To achieve personalized QoS prediction and reliable cloud service recommendation, in [230], the authors proposed a clustering-based and trust-aware approach. In [231], the authors improved the sparsity and new user problems in CF by clustering the personality factors on the STS (South Tyrol Suggests) dataset. The proposed model categorizes users based on their personality traits. In [232], the authors Fuzzy Side Information Clustering-Based Framework for Effective Recommendations on MovieLens and Yahoo datasets. The proposed model is based on side information and the Mahalanobis distance measure. In [233], the authors presented a probabilistic item embedding model that learns item representations from click data, as well as an EMB-MF model that combines the probabilistic item embedding and PMF for coupling the two models' item representations. In the study conducted by [234], RS with linked open data (RS-LOD) model was proposed. To address the issues of data sparsity and cold start, the authors developed a MF model with Linked Open Data (MF-LOD). In [235], the authors presented a hybrid recommendation approach with side information using a Gaussian mixture model and MF technology, where the improved cosine similarity formula is used to find users' neighbors, and initial ratings on unrated items are predicted. To reduce the problem of cold start and data sparsity, user ratings on items are converted into user preferences on item attributes. In [236], the authors proposed a social spatiotemporal probabilistic MF (SSTPMF) model that exploits POI similarity and user similarity and integrates different spaces in similarity modeling, including the social space, geographical space, and POI category space.

5.2. Remedies for challenges of CF-based on DL models

DL techniques are one of the trends in CF RS that are being used to provide appropriate treatments for the CF algorithm's drawbacks. In this section, we will investigate and explain studies that use DL to tackle issues with RS.

5.2.1. Remedies for issues with sparsity and cold start

One method for overcoming the sparsity problem in CF algorithms is to utilize DL techniques to turn the high-dimensional and sparse matrix of user and item into a lower-dimensional and denser set [76, 97, 104, 95, 237, 238, 239]. For example, in [95], the authors extrapolated the low-dimensional latent space of context features extracted from sensors using auto-encoders to handle the sparsity problems in those systems.

Using extra side information, the effects of data sparsity in the user-item rating matrix are mostly mitigated [240, 241]. Owing to the sparsity of the rating matrix and side information, latent features may be ineffective. As a result, DL approaches are used to generate high-level representations of the user-item rating matrix and side data, which are then combined with MF to solve sparsity and cold-start problems [102, 98, 103, 129, 89, 242, 100, 243]. For instance, in [103] the authors used marginalized DAE to incorporate MF-based CF and deeply learned features. In [102], the authors presented a probabilistic SDAE formulation, which they later extended to a relational SDAE (RSDAE) model. RSDAE conducts deep representation learning and relational learning concurrently in a systematic manner within a probabilistic framework to deal with sparsity issues by incorporating the extracted features into MF. In [98], introduced a hierarchical Bayesian version of the stacked denoising autoencoder DL model (SDAE). The next shows their CDL model, which tightly links deep representation learning for extra information with CF for the rating matrix, allowing two-way interaction. In [129], the authors suggested using a latent factor model for recommendation and predicting latent factors from music audio when consumption data could not be provided using the CNN method to address the cold-start issue of CF. In [106], the authors combined features extracted from extra information and timeSVD++ with SDAEs to address sparsity and cold-start issues. In [244], the authors used the extra information to address the scarcity and cold start issues with the blog recommendations. They incorporate features extracted from text and images, respectively, using the word2vec and CNNs approaches, into their proposed boosted inductive matrix completion method. In [130], the authors address the sparsity issue by incorporating latent factors extracted with CNNs into the MF approach via the latent factor model. In [105], the authors extracted hidden features from extra information using SDAE and integrated them into the pairwise ranking method's Bayesian framework to address the sparsity issue.

In some studies, side information is directly incorporated into the recommendation process to address sparsity and cold start issues, such as [104, 79, 243, 245, 246, 247, 108]. For example, the authors incorporated side information for every stack of the DAE method [104]. In [79], the authors tie Boltzmann machines to item-side information to deal with cold-start issues. In [243], the authors used a CNN method to extract hidden features from images to alleviate tag scarcity in tag-aware RS. Moreover, in [245], the authors proposed a model that trains item characteristics using implicit user reviews and textual side information using DL neural networks. They do not, however, consider the context or sequence of the sentences. Another method for responding to the cold-start problem is to create forecasts based on current browser action rather than browser history. In [121], the authors proposed an RNN model to produce predictions based on current user activities; in [248]the authors used an RNN approach to incorporate recent user sessions to deal with cold-start problems for new users in the current session. Moreover, in [249] the authors used 3D-CNNs to address sparsity issues in session-based RS by incorporating item-side information. In [250], the authors used a Bayesian inference approach combined with an RNN approach to deal with sparsity issues in session-based RS.

In the field of music recommendation, in [251], the authors proposed the classifier DL method, which is used during network training to deal with cold-start issues. Most sparsity and cold-start issues are dealt with by using DL methods for feature engineering to extract features from heterogeneous data sources.

5.2.2. Remedies for Scalability Problem

To deal with large-scale datasets, the researchers used deep neural networks to project the high-dimensional representations onto low-dimensional latent variables from user preferences and item ratings. Adapting DL approaches to scalability issues is the preferred method for system recommendation. [74, 78, 76, 252]. For example, in [252], the authors utilized deep neural networks based on multi-view to turn high-dimensional variables into lower-dimensional space, in [78], the authors used the Boltzmann method to extract hidden latent variables of items and users to generate predictions across large-scale datasets. Furthermore, to scale up their framework and train the network, they used methods for dimensional reduction, such as relevant attributes, the top-k most, k-means clustering, and local sensitive hashing of user attributes. Some authors used the RBM approach for dimensionality reduction. In [253], the authors handle large-scale datasets by proposing several methods, including parallel computing with shared memory, distributed computing, and method ensembles for modifying the RBM CF method. According to authors in [253], in terms of recommendation quality, parallel computing is more effective than other mechanisms for dealing with the huge of users and items. In [254], the authors used an auto-regressive model that included both users and items to handle the scalability problem.

5.2.3. Remedies for enhancing accuracy

In RS, which produces predictions, accuracy is one of the primary goals of using DL techniques. The authors used DL approaches to extract latent factors because they are effective at extracting hidden features. According to [74], when incorporating the RBM approach with the SVD method, it achieves good predictions, unlike Netflix's RS. The authors proposed the auto-encoder-based predictor AutoRec in [93]. Auto-Rec outperforms biased matrix factorization on the MovieLens and Netflix datasets, according to experiments. [246] applied NADE to the CF process on their proposed model to improve accuracy and performance. The experiments demonstrate that the proposed model outperforms existing algorithms such as LLORMA, On MovieLens, and Netflix datasets. By reconstructing the dense form of user preferences in [96], the authors proposed a CDAE model for the top-N recommendation. The experiments on several datasets, including MovieLens, Netflix, and Yelp, demonstrated that the best recommendations depend on the dataset and that non-linear functions improve recommendations. Consequently, according to the proposed model, CDAE outperforms the exciting top-N recommendation approaches in terms of accuracy. In their work [95], authors proposed a method to improve the accuracy of context-aware recommendation systems by utilizing auto-encoders and principal component analysis (PCA) to extract latent context features from sensor data. The experimental results demonstrate that the use of only latent contextual features with an auto-encoder yields the best results, particularly when both positive and negative feedback is incorporated into the model. An auto-encoder method was used to optimize the user-item rating matrix initialization step in MF by authors in [75] to obtain high-accuracy RS-based trust awareness. The experimental results demonstrate that it outperforms the existing models of RS-based trust awareness. DL algorithms are typically used in RS to obtain user and item features, generate a combined method of either user-based and item-based methods, or side information with preference information.

In [78], the authors used Boltzmann methods to incorporate latent factors of user behaviors with intra-user and intra-item correlations. SVD methods were used to compare user- and item-centric Boltzmann methods, as well as a couple of modeling of both user- and item-centric mechanisms. On the MovieLens dataset, the experimental results of the proposed model outperform SVD, particularly for the combined method with correlation. Moreover, in [102], the authors developed a hybrid tag-aware RS that combines SDAEs and side information. In addition, they created a probabilistic SDAE method for learning the relationship between items, followed by a relational SDAE method that combines layered representational learning and relational learning. On the CiteULike and MovieLens datasets, the proposed relational SDAE outperforms the existing tag-aware recommendation methods.

In [77], the authors used DL techniques in a group-based recommendation to improve accuracy efficiently. A DL approach was used to extract collective features from the individual features of group members, which were then combined with group preferences. Researchers have found that DL performs better than existing methods, according to experiments. Using DL to observe user preferences in real time enhances recommendation quality. For instance, in [123], the authors used RNNs to model the coevolution of user-item interactivity in user preferences to efficiently improve the accuracy of the predictions. This method outperforms existing models of user-item interactions on several datasets, including IPTV, Reddit, and Yelp. In [124], the authors converted the recommendation function into a sequence prediction problem and used the RNNs approach to efficiently improve the short-term prediction accuracy and item coverage in the CF algorithm. In terms of short-term prediction accuracy, according to the results of the experiments, the proposed model outperforms existing top-N recommendation models. For real-time recommendations based on current user preferences, in [119] the authors used RNNs methods. Models using feed-forward neural networks have been developed to simulate the CF approach by considering user purchase history. In [255], the authors proposed a hybrid collaborative recurrent auto-encoder based on implicit feedback to improve accuracy.

Studies have shown that DL approaches can efficiently improve the accuracy of CF because of their ability to extract hidden factors and combine information from different sources, which is a major strength. Since there are more ratings per item than users, this improvement was made.

5.3. The Trends and Perspective of Authors to Solve the Issues of CF-based RS

Recently, researchers have tended to solve the problems of the RS by integrating explicit and implicit feedback in many ways.

For instance, in the ML techniques, many studies have found that there is feedback between explicit and implicit feedback in a complementary relationship [176, 256, 257, 258], will use both at the same time, there is a great potential to improve the effectiveness of the recommended. In [176], the authors proposed whether the user has rated this behavior As implicit feedback, using the hidden factors of all items that users have rated to represent users, and using explicit feedback and implicit feedback for recommendation tasks at the same time, asymmetric SVD (asymmetric-SVD, referred to as ASVD) and SVD++ are proposed. In [204], the authors Combine the SVD++ [176], and in [258], the author optimizes the expected reciprocal rank (ERR) index, a personalized ranking model MERR-SVD++ with explicit and implicit feedback is proposed. In [257], the authors considered the heterogeneity of explicit and implicit feedback, that is, the explicit feedback is mostly numerical, while the implicit feedback is mostly binary. To eliminate the numerical difference between the two, both were converted to values between 0 and 1, and different weights were assigned to them, respectively. An MF model of Co-rating based on score prediction was proposed. However, the practice of converting the explicit feedback data into a value between 0 and 1 weakens Explicit feedback data reflects the ability of users to express their preferences but fails to take into account the important characteristics of explicit feedback data. DL methods have been gradually introduced into the RS to learn user preferences for items [256, 259, 260]. DL algorithms are applied to learn the intricate user-item mapping relationship between the latent factor representation and the matching function. For instance, in [259], the authors proposed three methods: generalized matrix factorization (GMF), multi-layer perceptron (MLP), and neural matrix factorization (NeuMF), which use different ways to model user-item interaction. In [256], the authors combined the advantages of representation learning and matching equation learning and proposed the DeepCF framework. Google's wide and DL RS are sets of two components: wide and deep. The wide part is a linear model that learns the simple interaction between features, can "memorize" the user's behavior, and recommends the content of interest to the user, but it requires a lot of time-consuming and laborious manual feature work. The deep part is a feed-forward deep neural network model [261]. Through the combination of these two parts and joint training, the two advantages of memory and generalization are finally obtained. In [262], the author proposed a coupled CF recommendation model based on non-IID; they train and integrate the explicit and implicit user-item couplings using DL. This model outperforms the above-mentioned work. However, models based on deep neural networks face serious overfitting problems, have high computational and storage complexity, and cannot adapt to large-scale data. [206] GPMF, a generalized probabilistic matrix factorization model for the recommendation, combines explicit and implicit feedback. [263] proposed SSAERec, a recommendation algorithm for rating prediction that incorporates a stacked sparse auto-encoder into matrix factorization and can learn efficient results from a user-item rating matrix. [264] proposed dynamic decay CF (DDCF) that captures user preference variations and incorporates the concept of dynamic time decay. It extends the concept of human brain memory to specify the level of a user's interests (i.e., immediate, short-term, or long-term).

In [265], the authors proposed a Social Promoter Score (SPS)-based recommendation. The proposed construct is two user-item interaction matrices with users' explicit SPS values and users' views of activities as implicit feedback. In [266], the authors proposed an end-to-end neural network model–GAMMA (Graph and Multi-view Memory Attention Mechanism). We aim to replace offline meta-path-based similarity or commuting matrix computation with a graph attention mechanism. As a solution to the challenge of social influence prediction, in [267], the authors proposed the Multi-view Influence Prediction Network (MvInf), a DL framework that blends multi-view learning with graph attention neural networks.

6. An evaluation of comprehensive literature in quantitative terms

Here, we have presented the whole literature with a quantitative assessment and evaluation of papers based on the time of publication, techniques, tasks, kind of publishing, and data sets.

6.1. Number of Publications Over the Years

The first assessment covers the total number of publications over the years. We have demonstrated the total number of publications over the years. Since 2010, there has been a growing trend toward RS, especially in the CF algorithm field. As a result, CF-based RS continues to be the most active and trending field for researchers. Given improvements in big data processing capabilities and the current trend of applying deep models to RS, it is reasonable to predict that collaboration between the two disciplines will continue to increase shortly (see Figure 3).

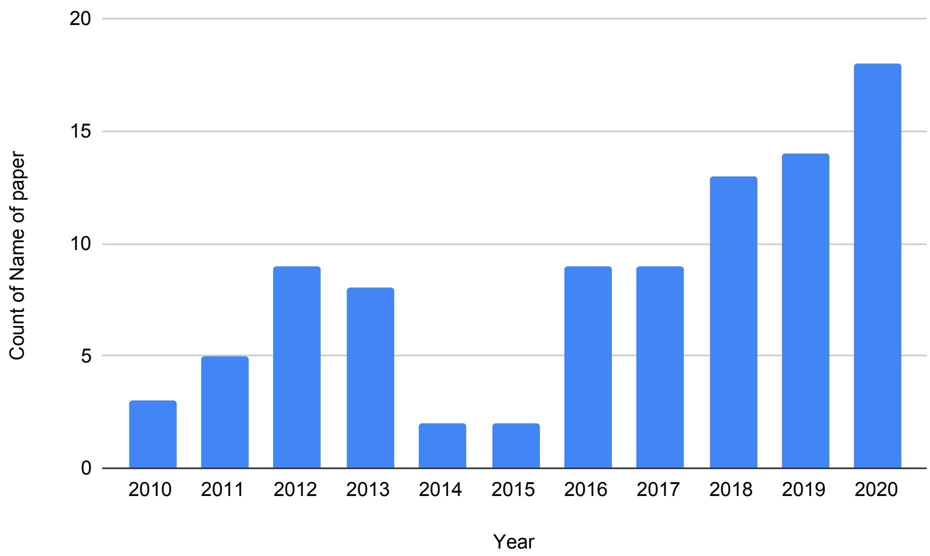


Figure 3: Number of publications over years in the area of Collaborative Filtering RS

6.2. Most applications of based RS

The second assessment covers the most applications of CF in the studies. We investigated in each paper the application that was used to handle the limitations of CF, and we found that among all the algorithms, the KNN and MF algorithms are the most widely used to deal with the limitations of CF RS, especially sparsity and scalability issues. We also found that hybrid algorithms are the most commonly used to handle the cold start problem, especially in the ML field. Regarding the DL algorithms, we found that the most common methods used to handle the limitations of CF are auto-encoder methods (see Table 1). Furthermore, these applications addressed the limitations of the CF algorithm by leveraging two types of data. Some models utilized user-item preferences, while others incorporated additional side information. Our analysis reveals that the majority of applications addressed these limitations by focusing on user-item preferences, as highlighted in Table 2.

Table 1: Summary of literature survey and an overview of collaborative filtering algorithms.

Begin of Table				
CF-based RS Algorithms	**Authors**	**Advantages**	**Limitations**	**Treatment**
KNN-based CF	[47, 158, 226, 229, 219, 220, 213, 215, 218, 192, 150, 268, 154, 164, 162, 153, 269, 144, 270, 271, 159, 140, 272, 151, 161, 156, 217, 165]	- Simplicity and interpretability. - Transparent recommendations based on similarity of users/items	- Scalability and High Computational Cost [273].	-Indexing methods such as KD trees and ball trees. - Approximate nearest neighbor algorithms such as locality-sensitive hashing and random projection[274].
MF-based CF	[53, 50, 155, 233, 211, 208, 236, 200, 207, 204, 205, 206, 201, 202, 203, 195, 197, 183, 192, 181, 182, 179, 180, 177, 178, 170, 176, 169, 174, 168, 34, 173, 175, 235, 171, 172, 224, 166, 164, 221]	- Handles sparse data - Enables personalized recommendations - Offers scalability for large datasets. - Addresses the cold start problem	- Assumes dense user-item interaction matrix, which is not in real-world with sparse data. - Face challenges when dealing with cold start problems [275, 276]	- Incorporate L1 or L2 regularization. - Utilizing side information to address the cold start problem [275, 276, 277].

Continuation of Table 1				
CF Algorithms	**Authors**	**Advantages**	**Limitations**	**Treatments**
Clustering-based CF	[60, 191, 143, 140, 65, 142, 72, 64, 166, 198, 146, 66, 147, 148, 199, 60, 152, 149, 145, 221, 219, 222, 229, 230, 231, 232]	- Offers the benefits of grouping similar users or items - Improves scalability by reducing computational complexity. - Uncovers new patterns in user-item interactions. - Enables accurate recommendations based on similarity.	- Determining the optimal number of clusters to form. - Finding the best representation of the data structure - Selecting clusters capturing diverse preferences [278, 279].	- Silhouette analysis - Elbow method - Gap statistic - Information criteria (e.g., BIC, AIC) [280].
Biclustering-based CF	[139, 140, 141, 142, 143, 154, 153, 155, 164, 165]	- Capturing local patterns - Considering user-item clusters instead of the entire dataset. - Improved accuracy by focusing on specific user-item subgroups.	- Identifying meaningful biclusters from large datasets can be computationally challenging. [281].	- Dimensionality reduction techniques such as MF or feature selection. - Incorporating domain knowledge or side information [278, 282].
Bayesian Classifier-based CF	[156, 162, 192, 195, 197, 212]	- Capturing the probability distribution of user-item preferences. - Handles sparse data by incorporating prior knowledge and smoothing techniques. - Allows for the inclusion of various user and item features.	- Struggle with new users or items that lack sufficient data for accurate probabilistic modeling [283].	- Capture dependencies between user-item interactions and features using advanced probabilistic models, such as Bayesian networks or Markov random fields [284].

Continuation of Table 1				
CF Algorithms	**Authors**	**Advantages**	**Limitations**	**Treatments**
Association Rule	[69, 70, 162, 186, 187, 188, 189, 190, 191, 194, 214, 213]	- Ability to uncover implicit relationships and dependencies among items.	- Sparsity of item associations. - Lack of user preferences - May struggle with new users or items that lack sufficient data for rule generation [285].	- Combining association rule-based CF with content-based filtering. - Setting appropriate thresholds for confidence and support measures - Incorporating user feedback [286].
Hybrid algorithms-based CF	[206, 205, 218, 219, 220, 202, 203, 200, 207, 208, 236, 209, 210, 211, 221, 228, 229, 230, 231, 232, 233, 234, 156, 158, 157, 162, 161, 153, 155, 154, 150, 166, 65, 64, 66, 148, 235]	- Handles the cold start problem, sparsity, and data scarcity more effectively. [206, 205, 287].	Integrating these techniques and balancing their strengths and weaknesses can be challenging.	Employed advanced ensemble learning methods such as weighted averaging, stacking, or hybridization at the feature level [209, 210, 288].

Continuation of Table 1				
CF Algorithms	**Authors**	**Advantages**	**Limitations**	**Treatments**
Fuzzy-based CF	[188, 152, 72, 148]	- Allows for representing and reasoning with uncertain and imprecise user preferences. - Handle noisy or incomplete data. - Provides a flexible framework for capturing user preferences.	- Involve complex computations due to fuzzy inference and similarity calculations. - The fuzzy nature of the approach can make it harder to explain the reasoning behind recommendations to users.	- Employing approximation methods, such as fuzzy clustering or rule reduction. - Use linguistic labels and explanations. [288, 289, 290].
Genetic-based CF	[70, 66, 190, 228]	- Optimize the recommendation model. - Handling complex relationships user-item interactions and dependencies. - Allows the system to adapt to evolving user preferences over time.	- Involves intensive computation. - Evolved recommendation models lack interpretability, hindering user explanations. - Tuning genetic algorithm parameters is challenging and time-consuming [291].	- Parallelization and optimization methods can reduce the computational complexity of genetic algorithms. - Post-processing techniques and rule extraction can improve interpretability of evolved models [70].

Continuation of Table 1				
CF Algorithms	**Authors**	**Advantages**	**Limitations**	**Treatments**
RBMs-based CF	[74, 75, 76, 78, 79, 76, 80, 81, 82, 83, 84, 85, 86, 87, 88]	- Capturing complex patterns and latent factors - Suitable for handling large datasets efficiently.	-Large datasets pose computational challenges and training difficulties due to the high number of parameters involved. [289].	- Scalability can be addressed by employing techniques like SGD and mini-batch training [292].
DBNs-based CF	[90, 91]	- Ability to model complex and non-linear relationships between users and items. - Automatic extract relevant features from raw user-item data without the need for manual feature engineering.	- Training and fine-tuning DBNs can be computationally expensive.	- Pre-training DBNs layer by layer and fine-tuning improves performance and reduces computational complexity [90].

Continuation of Table 1				
CF Algorithms	**Authors**	**Advantages**	**Limitations**	**Treatments**
Autoencoders-based CF	[92, 93, 75, 94, 95, 94, 96, 97, 98, 102, 103, 104, 105, 106, 106, 107, 108, 114, 109, 110, 112, 111, 113, 115, 116, 118, 293, 93, 103, 45]	- Capturing complex non-linear relationships improves recommendation accuracy. - Learning meaningful latent representations captures underlying patterns in user-item data. - Data sparsity is effectively handled by learning compact representations of users and items.	- Effectively capturing the sparsity and high-dimensionality of user-item interaction data is a challenge.	- Use dropout, regularization (L1/L2), or variational autoencoders to reduce overfitting and impose sparsity constraints. - Use methods like MF with autoencoders for dimensionality reduction and capturing meaningful latent representations. [45].
RNNs-based CF	[120, 119, 120, 121, 122, 123, 124, 125, 126, 127]	- Effectively model sequential user-item interactions, capturing temporal dependencies and patterns. - Handle variable-length sequences, accommodating diverse user behaviors and item sequences. - Learn personalized representations for users and items from their sequential interactions.	- RNNs may face challenges in handling new users or items with limited sequence data. - RNNs struggle with long-term dependencies, risking information loss. - Training RNNs is computationally intensive.	- Combining RNNs with other content filtering to address cold-start problem - Use Attention mechanisms - Use dropout or batch normalization can help prevent overfitting and improve generalization [294, 295, 296].

Continuation of Table 1				
CF Algorithms	**Authors**	**Advantages**	**Limitations**	**Treatments**
CNN-based CF	[129, 130, 131, 132, 297, 133, 134, 135, 115]	- Detect local patterns and features in user-item data - Invariant to the translation of features - Exploit parameter sharing, reducing the number of parameters and enabling efficient training and inference.	CNNs, designed for image and spatial data analysis, may be less suitable for modeling sequential or temporal information in user-item interaction sequences.	-By incorporating RNNs or attention mechanisms can effectively capture sequential and temporal dynamics [298, 299].
End of Table				

- Note: This table provides a summary of CF algorithms. For detailed discussions on the specific implementations and differences, please refer to the relevant sections of the manuscript:
  - Machine Learning-Based CF: Sections 4.3.1–4.3.5.
  - Deep Learning-Based CF: Sections 4.4.1–4.4.5. For an overall discussion, see Sections 5.1 and 5.2.

Table 2: Categorizing Publications of CF Algorithms based on Data Modeling

Begin of Table			
Data Modeling	**Authors**	**Advantages**	**Limitations**
User-Item Preferences	[47, 50, 165, 142, 217, 270, 272, 271, 141, 269, 144, 156, 157, 158, 159, 161, 154, 162, 153, 155, 166, 66, 72, 152, 60, 145, 146, 147, 148, 149, 150, 151, 300, 186, 187, 188, 189, 190, 191, 34, 168, 169, 170, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 192, 195, 183, 195, 194, 196, 197, 219, 220, 213, 214, 215, 216, 221, 223, 301, 225, 228, 224, 227]	- Focus solely on the interaction data, making them straightforward to implement. - Often readily available in many recommendation scenarios, enabling easy model training.	- May struggle with new users or items that have limited or no interaction data. - With limited data, accurately capturing user preferences and item characteristics can be challenging, leading to lower recommendation accuracy.
With Side Information	[235, 69, 268, 60, 151, 144, 164, 65, 64, 198, 218, 212, 202, 203, 204, 201, 205, 206, 200, 207, 208, 236, 209, 210, 211, 222, 226, 229, 230, 231, 232, 233, 234]	- Provides additional context about users and items, enabling more accurate and personalized recommendations. - Can help mitigate the cold-start problem by leveraging user or item attributes when interaction data is sparse.	- Obtaining and maintaining reliable side information for all users and items can be time-consuming and resource-intensive. - Choosing relevant and informative side information features can be challenging, and using irrelevant features may lead to noise or bias in recommendations.
End of Table			

6.3. Most Metrics used in CF

The third assessment covers the metrics of the CF (See Figure 4). Although a few researchers use custom and private metrics evaluation, the majority of experiments use metrics such as MAE and RMSE for regression problems and recall, precision, and F1 for classification problems. In fact, for evaluating RS, these metrics are overly simplistic. Unlike the leave-one-out evaluation to test the performance of item recommendation, which has been widely used in comparison models for implicit feedback recommendation using the Normalized Discounted Cumulative Gain (NDCG) and Hit Ratio @ 10 metrics.

6.4. Most Frequently Recommendation Task of CF

The fourth evaluation focuses on the recommendation task, which includes score prediction, recommendation list, recommendation set, and other tasks utilized in the research. We analyzed each paper's task and presented the distribution of used tasks as a pie chart (See Figure 5 and Table 3). Between all, 53% of the tasks are classified as "score predictions," and this category includes the tasks used in sharing with the Top-n recommendation list and Top-n recommendation set of the research. In the experiment, it is clear that the tasks of predicting scores and giving suggestions are the most popular. This is because this important information is taken into account.

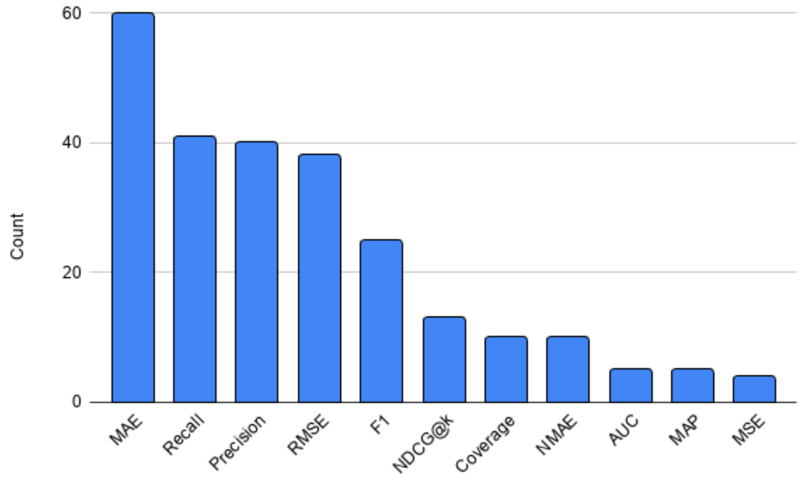


Figure 4: Most Frequently used Metrics

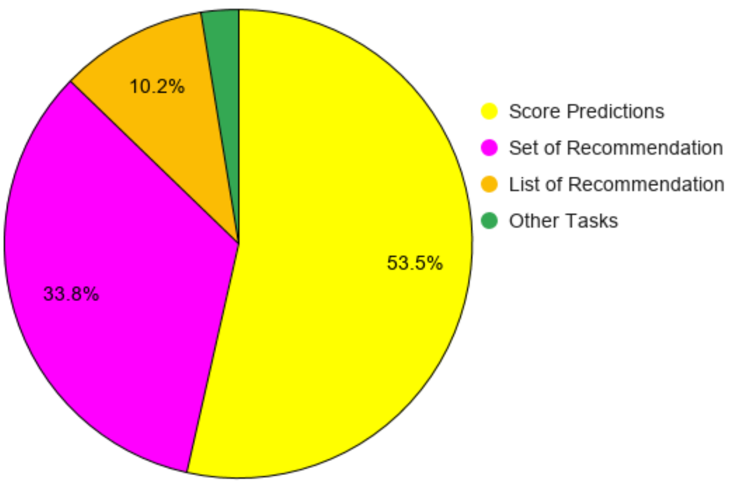


Figure 5: Most Frequently Recommendations Task

Table 3: Recommendation Task of Publications

Begin of Table				
Reco Task	**Authors**	**Metrics**	**Advantages**	**Limitations**
Score Predictions	[47, 50, 69, 217, 165, 151, 272, 142, 141, 270, 271, 157, 144, 158, 161, 162, 153, 154, 155, 164, 198, 146, 166, 66, 199, 147, 148, 149, 150, 175, 235, 34, 168, 169, 170, 173, 174, 176, 179, 180, 181, 182, 183, 195, 197, 219, 215, 218, 216, 202, 203, 204, 201, 205, 206, 200, 207, 208, 236, 210, 211, 221, 225, 224, 227, 228, 222, 226, 229, 230, 231, 232, 233, 234]	- RMSE - MAE - NMAE	- Provide a quantitative measure of the accuracy of predicted ratings.	- May not directly reflect the usefulness of recommendations for users. - Can be sensitive to outliers and biased towards well-predicted ratings.
Set of Recommendations	[162, 60, 230, 236, 206, 216, 202, 301, 214, 213, 212, 220, 219, 210, 218, 208, 195, 194, 188, 190, 192, 189, 177, 191, 34, 147, 146, 66, 187, 235, 151, 199, 145, 72, 198, 161, 152, 156, 269, 144, 64, 271, 65, 270, 154].	- Precision - Recall - F1 - HR - ARHR - ROC - AUC	- Focus on evaluating the effectiveness of generating a relevant set of recommendations.	- May not consider the quality of individual recommendations within the set. -Can be affected by the choice of the threshold for determining relevance.

Continuation of Table 3				
Reco Task	**Authors**	**Metrics**	**Advantages**	**Limitations**
List of Recommendations	[144, 162, 154, 186, 199, 300, 268, 177, 220, 214, 202, 204, 206, 208, 236, 223, 209]	- HL - NDCG - MAP - RBP - DCG	- Consider the ranking order of recommendations, reflecting the importance of the top-ranked items.	- Can be sensitive to the length of the recommendation list and heavily influenced by the top-ranked items. - May not capture user preferences beyond the top items.
Other Task	[159, 152, 199, 64]	- Diversity - Novelty - Stability	- Address additional goals like subject diversity, novelty, and stability of recommendations.	- These metrics may be more subjective and context-dependent. - There is no universally accepted standard for evaluating these aspects.
End of Table				

6.5. Most of Dataset in CF

The fifth evaluation focuses on the dataset used in the articles. We analyzed each paper's dataset and displayed the distribution of used datasets as a pie chart in Figure 6. According to the comprehensive survey of studies, the majority of the studies conducted their experiments on the MovieLens dataset.

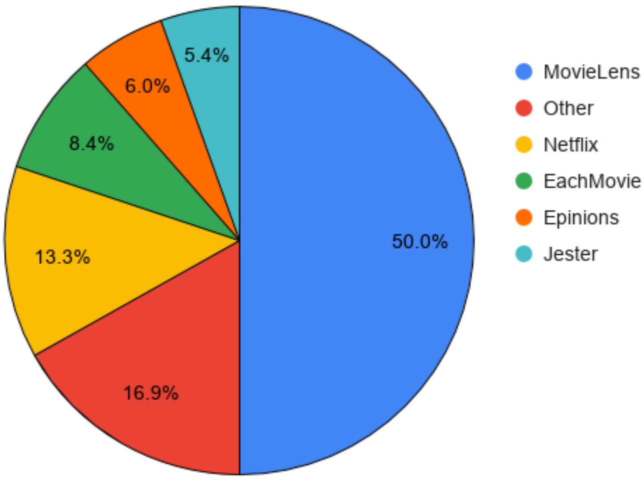


Figure 6: Most Frequently used datasets.

6.6. Categories of publications

Six, we examined the research based on how it was published. We received three primary categories of publications: journal articles, conference papers, chapter book papers, and dissertations. The total distribution is depicted in Figure 7. Because more than half of the research is published in journals, this is the most common kind of publication among the articles. We knew that we regarded the research articles published on arXiv to be research papers under the journal categorization.

7. Evolutionary Journey of CF-based RS

The evolutionary journey of collaborative filtering (CF) has shaped the field of recommendation systems. From its origins to its current state, CF has

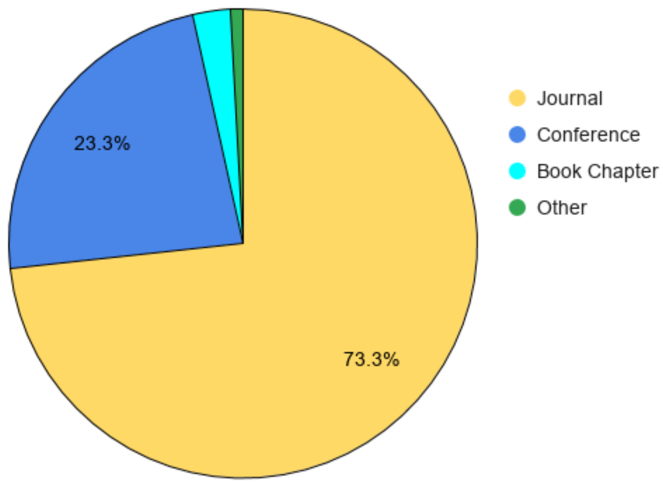


Figure 7: Distribution of Publications by their type

undergone significant development, enabling personalized recommendations. In this section, we explore CF's origins, development, and future directions, highlighting its transformative impact in the realm of personalized recommendations.

7.1. Origins of CF

CF originated in the late 1980s as a user-centric approach in recommendation systems. It emerged as a response to the limitations of content-based filtering methods. By considering the preferences and behaviors of similar users, CF provides personalized recommendations. Its origins marked a significant shift in the field and paved the way for its widespread adoption in various domains.

In the early stages of CF, two primary approaches were developed: user-based CF and item-based CF. Both user-based and item-based CF approaches [47] were early attempts to leverage the opinions and choices of users to provide personalized recommendations. These methods laid the foundation for further advancements and refinements in CF techniques [47].

MF techniques have been a significant development in CF, particularly in the early 2000s [52]. These techniques aim to improve recommendation accuracy by decomposing the user-item interaction matrix into lower-dimensional representations. MF techniques, including SVD [302] and NMF [167], have significantly improved recommendation quality by capturing underlying patterns and relationships between users and items. These techniques enable CF algorithms to generate accurate and personalized recommendations based on the decomposition of the user-item interaction matrix [52].

Neighborhood-based methods are an important development in CF that aim to provide recommendations based on the preferences of similar users or items. Neighborhood-based methods leverage the idea that users or items with similar preferences tend to have similar preferences in the future [47, 158, 226, 229, 219]. These methods provide personalized recommendations by considering the preferences of neighbors and utilizing their past behaviors. By identifying and incorporating the preferences of similar users or items, neighborhood-based methods enhance the accuracy and relevance of CF recommendations.

Hybrid approaches in CF have gained significant interest by combining multiple recommendation techniques to improve the accuracy and diversity of recommendations [206, 205, 218, 219, 220]. These approaches leverage the strengths of different methods and overcome the limitations of individual techniques. Two common hybrid approaches are content-based filtering and context-aware recommendation.

7.2. Development of CF-based RS

Deep learning techniques have made a significant impact on collaborative filtering by enabling the modeling of complex patterns and representations in user-item interactions. These techniques leverage deep neural networks to capture and analyze intricate relationships in large-scale datasets, leading to enhanced recommendation accuracy and performance.

One common application of deep learning in CF is through deep MF models. Autoencoders and Restricted Boltzmann Machines (RBMs) are examples of deep MF models. These models learn low-dimensional representations of users and items by reconstructing the user-item interaction matrix [74, 75, 76]. By encoding the latent factors in the data, deep matrix factorization models can generate more accurate recommendations.

Neural Collaborative Filtering (NCF) is another powerful deep learning-based approach [259]. NCF models use neural networks to directly learn user and item embeddings from the data. By combining collaborative filtering techniques with deep learning, NCF models can capture both user-item interactions and content-based information, resulting in more effective recommendations.

Recurrent Neural Networks (RNNs) have also been applied in collaborative filtering to model sequential patterns in user behavior. RNN-based models can capture temporal dependencies and evolving user preferences over time [120, 119]. By considering the sequential nature of user-item interactions, RNNs enable more personalized and adaptive recommendations

7.3. Future and Ongoing Research in CF-based RS

- Deep Learning in CF: Deep learning methods, such as neural networks and deep matrix factorization, continue to be an active research area in CF [303, 304, 305]. Researchers are exploring novel architectures, loss functions, and regularization techniques to improve recommendation accuracy and handle challenges like data sparsity, cold-start, and scalability.

- Hybrid Models [306]: Hybrid CF models that combine multiple recommendation techniques are being extensively studied. This includes integrating CF with content-based filtering, context-aware recommendation, reinforcement learning, or knowledge graphs. The aim is to leverage the complementary strengths of different approaches for more accurate and diverse recommendations.

- Explainable CF: Enhancing the interpretability and explainability of CF models is an ongoing research focus [307, 308]. Researchers are developing techniques to provide transparent explanations for CF recommendations, enabling users to understand the underlying factors that influence the suggestions.

- Transfer Learning in CF [309, 310]: Transfer learning, which transfers knowledge from one domain to another, is gaining attention in CF [311]. Researchers are investigating techniques to transfer knowledge across domains or tasks, improving recommendation performance in scenarios with limited data or addressing the cold-start problem.

- Privacy-Preserving CF [310, 311]: Addressing privacy concerns in CF is an important research area. Ongoing work focuses on developing privacy-preserving CF algorithms that protect sensitive user data while maintaining recommendation accuracy. Techniques like differential privacy, federated learning, and secure multiparty computation are explored to ensure privacy preservation.

- Reinforcement Learning in CF [278]: Reinforcement learning (RL) is being applied to CF to optimize long-term user engagement and satisfaction [312, 313]. RL algorithms are used to model the sequential decision-making process in recommendations, to maximize user rewards or utility.
- Contextual CF: Integrating contextual information, such as time, location, and social context, into CF models is an active research direction [314, 307]. Contextual CF models aim to provide more personalized and relevant recommendations by considering the specific context in which recommendations are made.
- Fairness and Bias in CF: Addressing fairness and bias issues in CF is gaining attention [315, 316]. Researchers are exploring techniques to mitigate biases in recommendations related to factors like gender, race, and socioeconomic background, ensuring fair and equitable recommendations for diverse user groups.
- Collaborative Filtering in New Domains: CF will expand its application to new domains, including healthcare [317], finance [318], education [319], and smart cities [320]. CF techniques will be tailored to specific domain requirements, considering domain-specific constraints, privacy regulations, and user preferences.

8. Conclusion

The prevalence of information filtering technologies, including Collaborative Filtering Recommender Systems (CF-RS), has grown in recent years due to the widespread availability of the Internet, the trend towards personalized experiences, and evolving user behaviors. Despite the effectiveness of current RS, they are still hindered by issues such as sparsity, accuracy, scalability, and cold start. To improve the quality of recommendations, advanced machine learning (ML) and deep learning (DL) techniques have been employed in RS development. This paper presents a comprehensive study of CF RS that utilizes ML and DL algorithms to provide valuable insights to new researchers who are interested in the field of RS. The study is divided into two sections: one for novice readers that covers the basic concepts of RS, and another for advanced readers that delves into the tasks, evaluations, and solutions of RS, emphasizing CF. The research is based on a thorough analysis of 307 systematic references that were selected based on the significance of the publication venue, including international conferences, journals, and book chapters, as well as the number of recent citations.

References

[1] Walter Carrer-Neto, María Luisa Hernández-Alcaraz, Rafael Valencia-García, and Francisco García-Sánchez. Social knowledge-based recommender system. application to the movies domain. *Expert Systems with applications*, 39(12):10990–11000, 2012.

[2] Pinata Winoto and Tiffany Y Tang. The role of user mood in movie recommendations. *Expert Systems with Applications*, 37(8):6086–6092, 2010.

[3] Seok Kee Lee, Yoon Ho Cho, and Soung Hie Kim. Collaborative filtering with ordinal scale-based implicit ratings for mobile music recommendations. *Information Sciences*, 180(11):2142–2155, 2010.

[4] Alexandros Nanopoulos, Dimitrios Rafailidis, Panagiotis Symeonidis, and Yannis Manolopoulos. Musicbox: Personalized music recommendation based on cubic analysis of social tags. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(2):407–412, 2009.

[5] Shulong Tan, Jiajun Bu, Chun Chen, Bin Xu, Can Wang, and Xiaofei He. Using rich social media information for music recommendation via hypergraph model. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 7(1):1–22, 2011.

[6] Zhiwen Yu, Xingshe Zhou, Yanbin Hao, and Jianhua Gu. Tv program recommendation for multiple viewers based on user profile merging. *User modeling and user-adapted interaction*, 16(1):63–82, 2006.

[7] Ana Belén Barragáns-Martínez, Enrique Costa-Montenegro, Juan C Burguillo, Marta Rey-López, Fernando A Mikic-Fonte, and Ana

Peleteiro. A hybrid content-based and item-based collaborative filtering approach to recommend tv programs enhanced with singular value decomposition. *Information Sciences*, 180(22):4290–4311, 2010.

[8] Edward Rolando Núñez-Valdéz, Juan Manuel Cueva Lovelle, Oscar Sanjuán Martínez, Vicente García-Díaz, Patricia Ordoñez De Pablos, and Carlos Enrique Montenegro Marín. Implicit feedback techniques on recommender systems applied to electronic books. *Computers in Human Behavior*, 28(4):1186–1193, 2012.

[9] Rubén González Crespo, Oscar Sanjuán Martínez, Juan Manuel Cueva Lovelle, B Cristina Pelayo García-Bustelo, José Emilio Labra Gayo, and Patricia Ordoñez De Pablos. Recommendation system based on user interaction data applied to intelligent electronic books. *Computers in human behavior*, 27(4):1445–1449, 2011.

[10] Jesus Serrano-Guerrero, Enrique Herrera-Viedma, Jose A Olivas, Andres Cerezo, and Francisco P Romero. A google wave-based fuzzy recommender system to disseminate information in university digital libraries 2.0. *Information Sciences*, 181(9):1503–1516, 2011.

[11] Carlos Porcel, Juan Manuel Moreno, and Enrique Herrera-Viedma. A multi-disciplinar recommender system to advice research resources in university digital libraries. *Expert Systems with Applications*, 36(10):12520–12528, 2009.

[12] Carlos Porcel and Enrique Herrera-Viedma. Dealing with incomplete information in a fuzzy linguistic recommender system to disseminate information in university digital libraries. *Knowledge-Based Systems*, 23(1):32–39, 2010.

[13] Carlos Porcel, A Tejeda-Lorente, MA Martínez, and Enrique Herrera-Viedma. A hybrid recommender system for the selective dissemination of research resources in a technology transfer office. *Information Sciences*, 184(1):1–19, 2012.

[14] Osmar R Zaíane. Building a recommender agent for e-learning systems. In *International Conference on Computers in Education, 2002. Proceedings.*, pages 55–59. IEEE, 2002.

[15] JESUS Bobadilla, Francisco Serradilla, Antonio Hernando, et al. Collaborative filtering adapted to recommender systems of e-learning. *Knowledge-Based Systems*, 22(4):261–265, 2009.

[16] Zan Huang, Daniel Zeng, and Hsinchun Chen. A comparison of collaborative-filtering recommendation algorithms for e-commerce. *IEEE Intelligent Systems*, 22(5):68–78, 2007.

[17] Jose Jesus Castro-Schez, Raul Miguel, David Vallejo, and Lorenzo Manuel López-López. A highly adaptive recommender system based on fuzzy logic for b2c e-commerce portals. *Expert Systems with Applications*, 38(3):2441–2454, 2011.

[18] Enrique Costa-Montenegro, Ana Belén Barragáns-Martínez, and Marta Rey-López. Which app? a recommender system of applications in markets: Implementation of the service for monitoring users' interaction. *Expert systems with applications*, 39(10):9367–9375, 2012.

[19] Kevin McNally, Michael P O'Mahony, Maurice Coyle, Peter Briggs, and Barry Smyth. A case study of collaboration and reputation in social web search. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(1):1–29, 2011.

[20] Rui Chen, Qingyi Hua, Yan-Shuo Chang, Bo Wang, Lei Zhang, and Xiangjie Kong. A survey of collaborative filtering-based recommender systems: From traditional methods to hybrid methods based on social networks. *IEEE Access*, 6:64301–64320, 2018.

[21] Mohammed FadhelAljunid and DH Manjaiah. A survey on recommendation systems for social media using big data analytics. *International Journal of Latest Trends in Engineering and Technology*, pages 48–58, 2017.

[22] Zhe Yang, Bing Wu, Kan Zheng, Xianbin Wang, and Lei Lei. A survey of collaborative filtering-based recommender systems for mobile internet applications. *IEEE Access*, 4:3273–3287, 2016.

[23] Newton Spolaôr, Huei Diana Lee, Weber Shoity Resende Takaki, Leandro Augusto Ensina, Claudio Saddy Rodrigues Coy, and Feng Chung Wu. A systematic review on content-based video retrieval. *Engineering Applications of Artificial Intelligence*, 90:103557, 2020.

[24] Mehdi Elahi, Francesco Ricci, and Neil Rubens. A survey of active learning in collaborative filtering recommender systems. *Computer Science Review*, 20:29–50, 2016.

[25] Zeynep Batmaz, Ali Yurekli, Alper Bilge, and Cihan Kaleli. A review on deep learning for recommender systems: challenges and remedies. *Artificial Intelligence Review*, 52(1):1–37, 2019.

[26] Yehuda Koren and Robert Bell. Advances in collaborative filtering. In *Recommender systems handbook*, pages 77–118. Springer, 2015.

[27] Christian Desrosiers and George Karypis. A comprehensive survey of neighborhood-based recommendation methods. In *Recommender systems handbook*, pages 107–144. Springer, 2011.

[28] Marko Balabanović and Yoav Shoham. Fab: content-based, collaborative recommendation. *Communications of the ACM*, 40(3):66–72, 1997.

[29] Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro. Content-based recommender systems: State of the art and trends. In *Recommender systems handbook*, pages 73–105. Springer, 2011.

[30] Marco De Gemmis, Pasquale Lops, Cataldo Musto, Fedelucio Narducci, and Giovanni Semeraro. Semantics-aware content-based recommender systems. In *Recommender systems handbook*, pages 119–159. Springer, 2015.

[31] Marco Degemmis, Pasquale Lops, and Giovanni Semeraro. A content-collaborative recommender that exploits wordnet-based user profiles for neighborhood formation. *User Modeling and User-Adapted Interaction*, 17(3):217–255, 2007.

[32] Robin Burke. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, 12(4):331–370, 2002.

[33] Kwon Y. Adomavicius G. Multi-criteria recommender systems. In *Recommender systems handbook*, pages 847–880. Springer, 2015.

[34] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Application of dimensionality reduction in recommender system-a case

study. Technical report, Minnesota Univ Minneapolis Dept of Computer Science, 2000.

[35] Joonseok Lee, Mingxuan Sun, and Guy Lebanon. A comparative study of collaborative filtering algorithms. *arXiv preprint arXiv:1205.3193*, 2012.

[36] Xiaoyuan Su and Taghi M Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009, 2009.

[37] Deuk Hee Park, Hyea Kyeong Kim, Il Young Choi, and Jae Kyeong Kim. A literature review and classification of recommender systems research. *Expert systems with applications*, 39(11):10059–10072, 2012.

[38] Linyuan Lü, Matúš Medo, Chi Ho Yeung, Yi-Cheng Zhang, Zi-Ke Zhang, and Tao Zhou. Recommender systems. *Physics reports*, 519(1):1–49, 2012.

[39] Dmitry Bogdanov, MartíN Haro, Ferdinand Fuhrmann, Anna Xambó, Emilia Gómez, and Perfecto Herrera. Semantic audio content-based music recommendation and visualization based on user preference examples. *Information Processing & Management*, 49(1):13–33, 2013.

[40] Lei Zheng. A survey and critique of deep learning on recommender systems. *no. September*, page 31, 2016.

[41] Basiliyos Tilahun Betru, Charles Awono Onana, and Bernabe Batchakui. Deep learning methods on recommender system: A survey of state-of-the-art. *International Journal of Computer Applications*, 162(10):17–22, 2017.

[42] Juntao Liu and Caihua Wu. Deep learning based recommendation: A survey. In *International Conference on Information Science and Applications*, pages 451–458. Springer, 2017.

[43] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys (CSUR)*, 52(1):1–38, 2019.

[44] Manqing Dong, Feng Yuan, Lina Yao, Xianzhi Wang, Xiwei Xu, and Liming Zhu. Trust in recommender systems: A deep learning perspective. *arXiv preprint arXiv:2004.03774*, 2020.

[45] Mohammed Fadhel Aljunid and Manjaiah Doddaghatta Huchaiah. Integratecf: Integrating explicit and implicit feedback based on deep learning collaborative filtering algorithm. *Expert Systems with Applications*, 207:117933, 2022.

[46] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pages 175–186, 1994.

[47] Badrul Munir Sarwar, George Karypis, Joseph A Konstan, John Riedl, et al. Item-based collaborative filtering recommendation algorithms. *Www*, 1:285–295, 2001.

[48] Markus Zanker and Daniel Ninaus. Knowledgeable explanations for recommender systems. In *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, volume 1, pages 657–660. IEEE, 2010.

[49] Sergio Cleger-Tamayo, Juan M Fernandez-Luna, and Juan F Huete. Explaining neighborhood-based recommendations. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 1063–1064, 2012.

[50] Jing Liu, Yu Jiang, Zechao Li, Xi Zhang, and Hanqing Lu. Domain-sensitive recommendation with user-item subgroup analysis. *IEEE Transactions on Knowledge and Data Engineering*, 28(4):939–950, 2015.

[51] Koji Miyahara and Michael J Pazzani. Collaborative filtering with the simple bayesian classifier. In *Pacific Rim International conference on artificial intelligence*, pages 679–689. Springer, 2000.

[52] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, (8):30–37, 2009.

[53] Mohammed Fadhel Aljunid and Manjaiah Dh. An efficient deep learning approach for collaborative filtering recommender system. *Procedia Computer Science*, 171:829–836, 2020.

[54] Yunhong Zhou, Dennis Wilkinson, Robert Schreiber, and Rong Pan. Large-scale parallel collaborative filtering for the netflix prize. In *International conference on algorithmic applications in management*, pages 337–348. Springer, 2008.

[55] Pentti Paatero. Least squares formulation of robust non-negative factor analysis. *Chemometrics and intelligent laboratory systems*, 37(1):23–35, 1997.

[56] Ajit P Singh and Geoffrey J Gordon. A unified view of matrix factorization models. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 358–373. Springer, 2008.

[57] Tong Zhang and Frank J Oles. Text categorization based on regularized linear classification methods. *Information retrieval*, 4(1):5–31, 2001.

[58] Daniel Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. *Advances in neural information processing systems*, 13:556–562, 2000.

[59] Jason DM Rennie and Nathan Srebro. Loss functions for preference levels: Regression with discrete ordered labels. In *Proceedings of the IJCAI multidisciplinary workshop on advances in preference handling*, volume 1. Kluwer Norwell, MA, 2005.

[60] Cong Tran, Jang-Young Kim, Won-Yong Shin, and Sang-Wook Kim. Clustering-based collaborative filtering using an incentivized/penalized user model. *IEEE Access*, 7:62115–62125, 2019.

[61] Greg Linden, Brent Smith, and Jeremy York. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, 7(1):76–80, 2003.

[62] David McSherry. Explaining the pros and cons of conclusions in cbr. In *European Conference on Case-Based Reasoning*, pages 317–330. Springer, 2004.

[63] Georgios Pitsilis, Xiangliang Zhang, and Wei Wang. Clustering recommenders in collaborative filtering using explicit trust information. In *IFIP International Conference on Trust Management*, pages 82–97. Springer, 2011.

[64] Mi Zhang and Neil Hurley. Novel item recommendation by user profile partitioning. In *2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, volume 1, pages 508–515. IEEE, 2009.

[65] Duen-Ren Liu and Ya-Yueh Shih. Hybrid approaches to product recommendation based on customer lifetime value and purchase preferences. *Journal of Systems and Software*, 77(2):181–191, 2005.

[66] Zan Wang, Xue Yu, Nan Feng, and Zhenhua Wang. An improved collaborative movie recommendation system using computational intelligence. *Journal of Visual Languages & Computing*, 25(6):667–675, 2014.

[67] Maryam Hosseini-Pozveh, Mohamadali Nematbakhsh, and Naser Movahhedinia. A multidimensional approach for context-aware recommendation in mobile commerce. *arXiv preprint arXiv:0908.0982*, 2009.

[68] Yoon Ho Cho, Jae Kyeong Kim, and Soung Hie Kim. A personalized recommender system based on web usage mining and decision tree induction. *Expert systems with Applications*, 23(3):329–342, 2002.

[69] Bamshad Mobasher, Xin Jin, and Yanzan Zhou. Semantically enhanced collaborative filtering on the web. In *European Web Mining Forum*, pages 57–76. Springer, 2003.

[70] Hossein Hatami Varzaneh, Behzad Soleimani Neysiani, Hassan Ziafat, and Nasim Soltani. Recommendation systems based on association rule mining for a target object by evolutionary algorithms. *Emerging Science Journal*, 2(2):100–107, 2018.

[71] Jesus Bobadilla, Fernando Ortega, Antonio Hernando, and Javier Alcalá. Improving collaborative filtering recommender system results and performance using genetic algorithms. *Knowledge-based systems*, 24(8):1310–1316, 2011.

[72] Jun Zhu, Lixin Han, Zhinan Gou, and Xiaofeng Yuan. A fuzzy clustering-based denoising model for evaluating uncertainty in collaborative filtering recommender systems. *Journal of the Association for Information Science and Technology*, 69(9):1109–1121, 2018.

[73] Bushra Alhijawi and Yousef Kilani. A collaborative filtering recommender system using genetic algorithm. *Information Processing & Management*, 57(6):102310, 2020.

[74] Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey Hinton. Restricted boltzmann machines for collaborative filtering. In *Proceedings of the 24th international conference on Machine learning*, pages 791–798. ACM, 2007.

[75] Shuiguang Deng, Longtao Huang, Guandong Xu, Xindong Wu, and Zhaohui Wu. On deep learning for trust-aware recommendations in social networks. *IEEE transactions on neural networks and learning systems*, 28(5):1164–1177, 2016.

[76] Kostadin Georgiev and Preslav Nakov. A non-iid framework for collaborative filtering with restricted boltzmann machines. In *International conference on machine learning*, pages 1148–1156, 2013.

[77] Liang Hu, Jian Cao, Guandong Xu, Longbing Cao, Zhiping Gu, and Wei Cao. Deep modeling of group preferences for group-based recommendation. In *AAAI*, volume 14, pages 1861–1867, 2014.

[78] Tran The Truyen, Dinh Q Phung, and Svetha Venkatesh. Ordinal boltzmann machines for collaborative filtering. *arXiv preprint arXiv:1205.2611*, 2012.

[79] Asela Gunawardana and Christopher Meek. Tied boltzmann machines for cold start recommendations. In *Proceedings of the 2008 ACM conference on Recommender systems*, pages 19–26, 2008.

[80] Hanene Ben Yedder, Umme Zakia, Aly Ahmed, and Ljiljana Trajković. Modeling prediction in recommender systems using restricted boltzmann machine. In *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2063–2068. IEEE, 2017.

[81] Fan Yang and Ying Lu. Restricted boltzmann machines for recommender systems with implicit feedback. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 4109–4113. IEEE, 2018.

[82] Dayal Kumar Behera, Madhabananda Das, Subhra Swetanisha, and Bighnaraj Naik. Collaborative filtering using restricted boltzmann machine and fuzzy c-means. In *Progress in Computing, Analytics and Networking*, pages 723–731. Springer, 2018.

[83] Abinash Pujahari and Dilip Singh Sisodia. Modeling side information in preference relation based restricted boltzmann machine for recommender systems. *Information Sciences*, 490:126–145, 2019.

[84] Naieme Hazrati, Bita Shams, and Saman Haratizadeh. Entity representation for pairwise collaborative ranking using restricted boltzmann machine. *Expert Systems with Applications*, 116:161–171, 2019.

[85] Zixiang Chen, Wanqi Ma, Wei Dai, Weike Pan, and Zhong Ming. Conditional restricted boltzmann machine for item recommendation. *Neurocomputing*, 385:269–277, 2020.

[86] Naieme Hazrati and Mehdi Elahi. Addressing the new item problem in video recommender systems by incorporation of visual features with restricted boltzmann machines. *Expert Systems*, page e12645, 2020.

[87] Mala Saraswat, Anil Dubey, Satyam Naidu, Rohit Vashisht, and Abhishek Singh. Web-based movie recommender system. In *Ambient Communications and Computer Systems*, pages 291–301. Springer, 2020.

[88] RJ Kuo and JT Chen. An application of differential evolution algorithm-based restricted boltzmann machine to recommendation systems. *Journal of Internet Technology*, 21(3):701–712, 2020.

[89] Xinxi Wang and Ye Wang. Improving content-based and hybrid music recommendation using deep learning. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 627–636. ACM, 2014.

[90] Kyo-Joong Oh, Won-Jo Lee, Chae-Gyun Lim, and Ho-Jin Choi. Personalized news recommendation using classified keywords to capture user preference. In *16th International Conference on Advanced Communication Technology*, pages 1283–1287. IEEE, 2014.

[91] Yifei Zhao, Jing Wang, and Feiyue Wang. Word embedding based retrieval model for similar cases recommendation. In *2015 Chinese Automation Congress (CAC)*, pages 2268–2272. IEEE, 2015.

[92] Yuanxin Ouyang, Wenqi Liu, Wenge Rong, and Zhang Xiong. Autoencoder-based collaborative filtering. In *International Conference on Neural Information Processing*, pages 284–291. Springer, 2014.

[93] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. Autorec: Autoencoders meet collaborative filtering. In *Proceedings of the 24th International Conference on World Wide Web*, pages 111–112. ACM, 2015.

[94] Fuzhen Zhuang, Zhiqiang Zhang, Mingda Qian, Chuan Shi, Xing Xie, and Qing He. Representation learning via dual-autoencoder for recommendation. *Neural Networks*, 90:83–89, 2017.

[95] Moshe Unger, Ariel Bar, Bracha Shapira, and Lior Rokach. Towards latent context-aware recommendation systems. *Knowledge-Based Systems*, 104:165–178, 2016.

[96] Yao Wu, Christopher DuBois, Alice X Zheng, and Martin Ester. Collaborative denoising auto-encoders for top-n recommender systems. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, pages 153–162. ACM, 2016.

[97] Florian Strub and Jeremie Mary. Collaborative filtering with stacked denoising autoencoders and sparse inputs. In *NIPS workshop on machine learning for eCommerce*, 2015.

[98] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1235–1244. ACM, 2015.

[99] Hong-Jian Xue, Xinyu Dai, Jianbing Zhang, Shujian Huang, and Jiajun Chen. Deep matrix factorization models for recommender systems. In *IJCAI*, pages 3203–3209, 2017.

[100] Shuai Zhang, Lina Yao, and Xiwei Xu. Autosvd++: An efficient hybrid collaborative filtering model via contractive auto-encoders. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 957–960. ACM, 2017.

[101] Wenjie Yan, Dong Wang, Mengjing Cao, and Jing Liu. Deep auto encoder model with convolutional text networks for video recommendation. *IEEE Access*, 7:40333–40346, 2019.

[102] Hao Wang, Xingjian Shi, and Dit-Yan Yeung. Relational stacked denoising autoencoder for tag recommendation. In *Twenty-ninth AAAI conference on artificial intelligence*. Citeseer, 2015.

[103] Sheng Li, Jaya Kawale, and Yun Fu. Deep collaborative filtering via marginalized denoising auto-encoder. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 811–820. ACM, 2015.

[104] Florian Strub, Romaric Gaudel, and Jérémie Mary. Hybrid recommender system based on autoencoders. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, pages 11–16. ACM, 2016.

[105] Haochao Ying, Liang Chen, Yuwen Xiong, and Jian Wu. Collaborative deep ranking: A hybrid pair-wise recommendation algorithm with implicit feedback. In *Pacific-asia conference on knowledge discovery and data mining*, pages 555–567. Springer, 2016.

[106] Jian Wei, Jianhua He, Kai Chen, Yi Zhou, and Zuoyin Tang. Collaborative filtering and deep learning based recommendation system for cold start items. *Expert Systems with Applications*, 69:29–39, 2017.

[107] Julio Barbieri, Leandro GM Alvim, Filipe Braida, and Geraldo Zimbrão. Autoencoders and recommender systems: Cofils approach. *Expert Systems with Applications*, 89:81–90, 2017.

[108] Xiaopeng Li and James She. Collaborative variational autoencoder for recommender systems. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 305–314, 2017.

[109] Giannis Karamanolakis, Kevin Raji Cherian, Ananth Ravi Narayan, Jie Yuan, Da Tang, and Tony Jebara. Item recommendation with variational autoencoders and heterogeneous priors. In *Proceedings of the 3rd Workshop on Deep Learning for Recommender Systems*, pages 10–14, 2018.

[110] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 world wide web conference*, pages 689–698, 2018.

[111] Yogesh Jhamb, Travis Ebesu, and Yi Fang. Attentive contextual denoising autoencoder for recommendation. In *Proceedings of the 2018 ACM SIGIR International Conference on Theory of Information Retrieval*, pages 27–34, 2018.

[112] Ming He, Qian Meng, and Shaozong Zhang. Collaborative additional variational autoencoder for top-n recommender systems. *IEEE Access*, 7:5707–5713, 2019.

[113] Yiteng Pan, Fazhi He, and Haiping Yu. A novel enhanced collaborative autoencoder with knowledge distillation for top-n recommender systems. *Neurocomputing*, 332:137–148, 2019.

[114] Harald Steck. Embarrassingly shallow autoencoders for sparse data. In *The World Wide Web Conference*, pages 3251–3257, 2019.

[115] Mohammed Fadhel Aljunid and Manjaiah Doddaghatta Huchaiah. Multi-model deep learning approach for collaborative filtering recommendation system. *CAAI Transactions on Intelligence Technology*, 5(4):268–275, 2020.

[116] Yiteng Pan, Fazhi He, and Haiping Yu. Learning social representations with deep autoencoder for recommender system. *World Wide Web*, 23(4):2259–2279, 2020.

[117] Jiajia Jiang, Weiling Li, Ani Dong, Quanhui Gou, and Xin Luo. A fast deep autoencoder for high-dimensional and sparse matrices in recommender systems. *Neurocomputing*, 412:381–391, 2020.

[118] Qusai Shambour. A deep learning based algorithm for multi-criteria recommender systems. *Knowledge-Based Systems*, 211:106545, 2021.

[119] Sai Wu, Weichao Ren, Chengchao Yu, Gang Chen, Dongxiang Zhang, and Jingbo Zhu. Personal recommendation using deep recurrent neural networks in netease. In *2016 IEEE 32nd international conference on data engineering (ICDE)*, pages 1218–1229. IEEE, 2016.

[120] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*, 2015.

[121] Yong Kiam Tan, Xinxing Xu, and Yong Liu. Improved recurrent neural networks for session-based recommendations. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, pages 17–22, 2016.

[122] Young-Jun Ko, Lucas Maystre, and Matthias Grossglauser. Collaborative recurrent neural networks for dynamic recommender systems. In *Asian Conference on Machine Learning*, pages 366–381. PMLR, 2016.

[123] Hanjun Dai, Yichen Wang, Rakshit Trivedi, and Le Song. Deep coevolutionary network: Embedding user and item features for recommendation. *arXiv preprint arXiv:1609.03675*, 2016.

[124] Robin Devooght and Hugues Bersini. Long and short-term recommendations with recurrent neural networks. In *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*, pages 13–21, 2017.

[125] Tu Minh Phuong, Tran Cong Thanh, and Ngo Xuan Bach. Neural session-aware recommendation. *IEEE Access*, 7:86884–86896, 2019.

[126] Yuanbo Xu, Yongjian Yang, Jiayu Han, En Wang, Jingci Ming, and Hui Xiong. Slanderous user detection with modified recurrent neural networks in recommender system. *Information Sciences*, 505:265–281, 2019.

[127] Kiewan Villatel, Elena Smirnova, Jérémie Mary, and Philippe Preux. Recurrent neural networks for long and short-term sequential recommendation. *arXiv preprint arXiv:1807.09142*, 2018.

[128] Meshal Alfarhood and Jianlin Cheng. Deephcf: A deep learning based hybrid collaborative filtering approach for recommendation systems. In

2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), pages 89–96. IEEE, 2018.

[129] Aaron Van den Oord, Sander Dieleman, and Benjamin Schrauwen. Deep content-based music recommendation. In *Advances in neural information processing systems*, pages 2643–2651, 2013.

[130] Xiaoxuan Shen, Baolin Yi, Zhaoli Zhang, Jiangbo Shu, and Hai Liu. Automatic recommendation technology for learning resources with convolutional neural network. In *2016 International Symposium on Educational Technology (ISET)*, pages 30–34. IEEE, 2016.

[131] Jiang Zhou, Rami Albatal, and Cathal Gurrin. Applying visual user interest profiles for recommendation and personalisation. In *International Conference on Multimedia Modeling*, pages 361–366. Springer, 2016.

[132] Chenyi Lei, Dong Liu, Weiping Li, Zheng-Jun Zha, and Houqiang Li. Comparative deep learning of hybrid representations for image recommendations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2545–2553, 2016.

[133] Donghyun Kim, Chanyoung Park, Jinoh Oh, Sungyoung Lee, and Hwanjo Yu. Convolutional matrix factorization for document context-aware recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 233–240. ACM, 2016.

[134] Serkan Kiranyaz, Turker Ince, Ridha Hamila, and Moncef Gabbouj. Convolutional neural networks for patient-specific ecg classification. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 2608–2611. IEEE, 2015.

[135] Serkan Kiranyaz, Onur Avci, Osama Abdeljaber, Turker Ince, Moncef Gabbouj, and Daniel J Inman. 1d convolutional neural networks and applications: A survey. *arXiv preprint arXiv:1905.03554*, 2019.

[136] J Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. Collaborative filtering recommender systems. In *The adaptive web*, pages 291–324. Springer, 2007.

[137] FO Isinkaye, YO Folajimi, and Bolande Adefowoke Ojokoh. Recommendation systems: Principles, methods and evaluation. *Egyptian informatics journal*, 16(3):261–273, 2015.

[138] Mohammed Fadhel Aljunid and DH Manjaiah. A survey on recommendation systems for social media using big data analytics. *International Journal of Latest Trends in Engineering and Technology, Special Issue (SACAIM 2017)*, pages 48–58.

[139] Jonathan L Herlocker, Joseph A Konstan, Al Borchers, and John Riedl. An algorithmic framework for performing collaborative filtering. In *22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 1999*, pages 230–237. Association for Computing Machinery, Inc, 1999.

[140] Badrul M Sarwar, George Karypis, Joseph Konstan, and John Riedl. Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering. In *Proceedings of the fifth international conference on computer and information technology*, volume 1, pages 291–324, 2002.

[141] Gui-Rong Xue, Chenxi Lin, Qiang Yang, WenSi Xi, Hua-Jun Zeng, Yong Yu, and Zheng Chen. Scalable collaborative filtering using cluster-based smoothing. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 114–121, 2005.

[142] Zhong Yao and Quang Zhang. Item-based clustering collaborative filtering algorithm under high-dimensional sparse data. In *2009 International Joint Conference on Computational Sciences and Optimization*, volume 1, pages 787–790. IEEE, 2009.

[143] Yi Cai, Ho-fung Leung, Qing Li, Huaqing Min, Jie Tang, and Juanzi Li. Typicality-based collaborative filtering recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 26(3):766–779, 2013.

[144] Soojung Lee. Using entropy for similarity measures in collaborative filtering. *Journal of Ambient Intelligence and Humanized Computing*, 11(1):363–374, 2020.

[145] N Sivaramakrishnan, V Subramaniyaswamy, Logesh Ravi, V Vijayakumar, Xiao-Zhi Gao, and SL Rakshana Sri. An effective user clustering-based collaborative filtering recommender system with grey wolf optimisation. *International Journal of Bio-Inspired Computation*, 16(1):44–55, 2020.

[146] Jia Zhang, Yaojin Lin, Menglei Lin, and Jinghua Liu. An effective collaborative filtering algorithm based on user preference clustering. *Applied Intelligence*, 45(2):230–240, 2016.

[147] Jiangzhou Deng, Junpeng Guo, and Yong Wang. A novel k-medoids clustering recommendation algorithm based on probability distribution for collaborative filtering. *Knowledge-Based Systems*, 175:96–106, 2019.

[148] Rahul Katarya and Om Prakash Verma. A collaborative recommender system enhanced with particle swarm optimization technique. *Multimedia Tools and Applications*, 75(15):9225–9239, 2016.

[149] Ricardo Erikson Veras De Sena Rosa, Felipe Augusto Souza Guimarães, Rafael da Silva Mendonça, and Vicente Ferreira de Lucena. Improving prediction accuracy in neighborhood-based collaborative filtering by using local similarity. *IEEE Access*, 8:142795–142809, 2020.

[150] Patricia Ortal and Masato Edahiro. Switching hybrid method based on user similarity and global statistics for collaborative filtering. *IEEE Access*, 8:213401–213415, 2020.

[151] Sanjaya Kumar Panda, Sourav Kumar Bhoi, and Munesh Singh. A collaborative filtering recommendation algorithm based on normalization approach. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–23, 2020.

[152] R Logesh, V Subramaniyaswamy, D Malathi, N Sivaramakrishnan, and V Vijayakumar. Enhancing recommendation stability of collaborative filtering recommender system through bio-inspired clustering ensemble method. *Neural Computing and Applications*, 32(7):2141–2164, 2020.

[153] Surya Kant and Tripti Mahara. Merging user and item based collaborative filtering to alleviate data sparsity. *International Journal of System Assurance Engineering and Management*, 9(1):173–179, 2018.

[154] Monika Singh and Monica Mehrotra. Impact of biclustering on the performance of biclustering based collaborative filtering. *Expert Systems With Applications*, 113:443–456, 2018.

[155] Surya Kant and Tripti Mahara. Nearest biclusters collaborative filtering framework with fusion. *Journal of Computational Science*, 25:204–212, 2018.

[156] Koji Miyahara and Michael J Pazzani. Improvement of collaborative filtering with the simple bayesian classifier. *Information Processing Society of Japan*, 43(11), 2002.

[157] Rong Hu and Yansheng Lu. A hybrid user and item-based collaborative filtering with smoothing on sparse data. In *16th International Conference on Artificial Reality and Telexistence–Workshops (ICAT'06)*, pages 184–189. IEEE, 2006.

[158] Akihiro Yamashita, Hidenori Kawamura, and Keiji Suzuki. Adaptive fusion method for user-based and item-based collaborative filtering. *Advances in Complex Systems*, 14(02):133–149, 2011.

[159] Jing Wang and Jian Yin. Combining user-based and item-based collaborative filtering techniques to improve recommendation diversity. In *2013 6th International Conference on Biomedical Engineering and Informatics*, pages 661–665. IEEE, 2013.

[160] Priyank Thakkar, Krunal Varma, and Vijay Ukani. Outcome fusion-based approaches for user-based and item-based collaborative filtering. In *International Conference on Information and Communication Technology for Intelligent Systems*, pages 127–135. Springer, 2017.

[161] Manel Slokom and Raouia Ayachi. A hybrid user and item based collaborative filtering approach by possibilistic similarity fusion. In *Advances in Combining Intelligent Methods*, pages 125–147. Springer, 2017.

[162] Priscila Valdiviezo-Diaz, Fernando Ortega, Eduardo Cobos, and Raúl Lara-Cabrera. A collaborative filtering approach based on naïve bayes classifier. *IEEE Access*, 7:108581–108592, 2019.

[163] Yilong Shi, Hong Lin, and Yuqiang Li. Iu-pmf: Probabilistic matrix factorization model fused with item similarity and user similarity. In *International Conference on Cloud Computing and Security*, pages 747–758. Springer, 2017.

[164] Chenjiao Feng, Jiye Liang, Peng Song, and Zhiqiang Wang. A fusion collaborative filtering method for sparse data in recommender systems. *Information Sciences*, 521:365–379, 2020.

[165] Dawei Wang, Yuehwern Yih, and Mario Ventresca. Improving neighbor-based collaborative filtering by using a hybrid similarity measurement. *Expert Systems with Applications*, 160:113651, 2020.

[166] Hafed Zarzour, Ziad Al-Sharif, Mahmoud Al-Ayyoub, and Yaser Jararweh. A new collaborative filtering recommendation algorithm based on dimensionality reduction and clustering techniques. In *2018 9th International Conference on Information and Communication Systems (ICICS)*, pages 102–106. IEEE, 2018.

[167] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *nature*, 401(6755):788–791, 1999.

[168] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Incremental singular value decomposition algorithms for highly scalable recommender systems. In *Fifth international conference on computer and information science*, volume 1, pages 27–8. Citeseer, 2002.

[169] Nathan Srebro and Tommi Jaakkola. Weighted low-rank approximations. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 720–727, 2003.

[170] Kai Yu, Shenghuo Zhu, John Lafferty, and Yihong Gong. Fast nonparametric matrix factorization for large-scale collaborative filtering. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 211–218. ACM, 2009.

[171] Emmanuel J Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717, 2009.

[172] Raghunandan H Keshavan, Andrea Montanari, and Sewoong Oh. Matrix completion from a few entries. *IEEE transactions on information theory*, 56(6):2980–2998, 2010.

[173] Andriy Mnih and Russ R Salakhutdinov. Probabilistic matrix factorization. *Advances in neural information processing systems*, 20:1257–1264, 2007.

[174] Ruslan Salakhutdinov and Andriy Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *Proceedings of the 25th international conference on Machine learning*, pages 880–887, 2008.

[175] Xin Luo, Yunni Xia, and Qingsheng Zhu. Incremental collaborative filtering recommender based on regularized matrix factorization. *Knowledge-Based Systems*, 27:271–280, 2012.

[176] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434. ACM, 2008.

[177] Gai Li and Weihua Ou. Pairwise probabilistic matrix factorization for implicit feedback collaborative filtering. *Neurocomputing*, 204:17–25, 2016.

[178] Yilmaz Ar. An initialization method for the latent vectors in probabilistic matrix factorization for sparse datasets. *Evolutionary Intelligence*, pages 1–13, 2019.

[179] Nathan Srebro, Jason Rennie, and Tommi Jaakkola. Maximum-margin matrix factorization. *Advances in neural information processing systems*, 17:1329–1336, 2004.

[180] Dennis DeCoste. Collaborative prediction using ensembles of maximum margin matrix factorizations. In *Proceedings of the 23rd international conference on Machine learning*, pages 249–256, 2006.

[181] Minjie Xu, Jun Zhu, and Bo Zhang. Nonparametric max-margin matrix factorization for collaborative prediction. In *Advances in Neural Information Processing Systems*, pages 64–72, 2012.

[182] Minjie Xu, Jun Zhu, and Bo Zhang. Fast max-margin matrix factorization with data augmentation. In *International Conference on Machine Learning*, pages 978–986, 2013.

[183] Vikas Kumar, Arun K Pujari, Sandeep Kumar Sahu, Venkateswara Rao Kagita, and Vineet Padmanabhan. Collaborative filtering using multiple binary maximum margin matrix factorizations. *Information Sciences*, 380:1–11, 2017.

[184] Jesús Bobadilla, Fernando Ortega, Antonio Hernando, and Abraham Gutiérrez. Recommender systems survey. *Knowledge-based systems*, 46:109–132, 2013.

[185] Ravi Nahta, Yogesh Kumar Meena, Dinesh Gopalani, and Ganpat Singh Chauhan. Embedding metadata using deep collaborative filtering to address the cold start problem for the rating prediction task. *Multimedia Tools and Applications*, pages 1–29, 2021.

[186] Weiyang Lin, Sergio A Alvarez, and Carolina Ruiz. Efficient adaptive-support association rule mining for recommender systems. *Data mining and knowledge discovery*, 6(1):83–105, 2002.

[187] Choonho Kim and Juntae Kim. A recommendation algorithm using multi-level association rules. In *Proceedings IEEE/WIC International Conference on Web Intelligence (WI 2003)*, pages 524–527. IEEE, 2003.

[188] Cane Wing-ki Leung, Stephen Chi-fai Chan, and Fu-lai Chung. A collaborative filtering framework based on fuzzy association rules and multiple-level similarity. *Knowledge and Information Systems*, 10(3):357–381, 2006.

[189] Shweta Tyagi and Kamal K Bharadwaj. Enhancing collaborative filtering recommendations by utilizing multi-objective particle swarm optimization embedded association rule mining. *Swarm and Evolutionary Computation*, 13:1–12, 2013.

[190] Behzad Soleimani Neysiani, Nasim Soltani, Reza Mofidi, and Mohammad Hossein Nadimi-Shahraki. Improve performance of association rule-based collaborative filtering recommendation systems using genetic algorithm. *Int. J. Inf Technol. Comput. Sci*, 2:48–55, 2019.

[191] Maryam Khanian Najafabadi, Mohd Naz'ri Mahrin, Suriayati Chuprat, and Haslina Md Sarkan. Improving the accuracy of collaborative filtering recommendations using clustering and association rules mining on implicit data. *Computers in Human Behavior*, 67:113–128, 2017.

[192] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618*, 2012.

[193] Tong Wang, Cynthia Rudin, Finale Doshi-Velez, Yimin Liu, Erica Klampfl, and Perry MacNeille. A bayesian framework for learning rule sets for interpretable classification. *The Journal of Machine Learning Research*, 18(1):2357–2393, 2017.

[194] Zhiyu Min and Dahua Lin. Probabilistic ensemble of collaborative filters. *arXiv preprint arXiv:1808.03298*, 2018.

[195] Guangyong Chen, Fengyuan Zhu, and Pheng Ann Heng. Large-scale bayesian probabilistic matrix factorization with memo-free distributed variational inference. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 12(3):1–24, 2018.

[196] Jingwen Yu, Zhanwei Xuan, Xiang Feng, Quan Zou, and Lei Wang. A novel collaborative filtering model for lncrna-disease association prediction based on the naïve bayesian classifier. *BMC bioinformatics*, 20(1):396, 2019.

[197] Antonio Hernando, Jesús Bobadilla, and Fernando Ortega. A non negative matrix factorization for collaborative filtering recommender systems based on a bayesian probabilistic model. *Knowledge-Based Systems*, 97:188–202, 2016.

[198] Surya Kant, Tripti Mahara, Vinay Kumar Jain, Deepak Kumar Jain, and Arun Kumar Sangaiah. Leaderrank based k-means clustering initialization method for collaborative filtering. *Computers & Electrical Engineering*, 69:598–609, 2018.

[199] Zhihua Cui, Xianghua Xu, Fei Xue, Xingjuan Cai, Yang Cao, Wensheng Zhang, and Jinjun Chen. Personalized recommendation system based on collaborative filtering for iot scenarios. *IEEE Transactions on Services Computing*, 2020.

[200] Hanhuai Shan and Arindam Banerjee. Generalized probabilistic matrix factorizations for collaborative filtering. In *2010 IEEE International Conference on Data Mining*, pages 1025–1030. IEEE, 2010.

[201] Guang Ling, Michael R Lyu, and Irwin King. Ratings meet reviews, a combined approach to recommend. In *Proceedings of the 8th ACM Conference on Recommender systems*, pages 105–112, 2014.

[202] Shulong Chen and Yuxing Peng. Matrix factorization for recommendation with explicit and implicit feedback. *Knowledge-Based Systems*, 158:109–117, 2018.

[203] Yonghong Yu, Can Wang, Hao Wang, and Yang Gao. Attributes coupling based matrix factorization for item recommendation. *Applied Intelligence*, 46(3):521–533, 2017.

[204] Gai Li and Qiang Chen. Exploiting explicit and implicit feedback for personalized ranking. *Mathematical Problems in Engineering*, 2016, 2016.

[205] Supriyo Mandal and Abyayananda Maiti. Explicit feedbacks meet with implicit feedbacks: a combined approach for recommendation system. In *International Conference on Complex Networks and their Applications*, pages 169–181. Springer, 2018.

[206] Supriyo Mandal and Abyayananda Maiti. Explicit feedback meet with implicit feedback in gpmf: a generalized probabilistic matrix factorization model for recommendation. *Applied Intelligence*, pages 1–24, 2020.

[207] Andrey Babkin. Incorporating side information into robust matrix factorization with bayesian quantile regression. *Statistics & Probability Letters*, 165:108847, 2020.

[208] Lei Chen, Zhiang Wu, Jie Cao, Guixiang Zhu, and Yong Ge. Travel recommendation via fusing multi-auxiliary information into matrix factorization. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(2):1–24, 2020.

[209] Abinash Pujahari and Dilip Singh Sisodia. Pair-wise preference relation based probabilistic matrix factorization for collaborative filtering in recommender system. *Knowledge-Based Systems*, page 105798, 2020.

[210] Mehmet Aktukmak, Yasin Yilmaz, and Ismail Uysal. A probabilistic framework to incorporate mixed-data type features: Matrix factorization with multimodal side information. *Neurocomputing*, 367:164–175, 2019.

[211] Xueming Qian, He Feng, Guoshuai Zhao, and Tao Mei. Personalized recommendation combining user interest and social circle. *IEEE transactions on knowledge and data engineering*, 26(7):1763–1777, 2013.

[212] Andrew I Schein, Alexandrin Popescul, Lyle H Ungar, and David M Pennock. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 253–260, 2002.

[213] Cane Wing-ki Leung, Stephen Chi-fai Chan, and Fu-lai Chung. An empirical study of a cross-level association rule mining approach to cold-start recommendations. *Knowledge-Based Systems*, 21(7):515–529, 2008.

[214] Timur Osadchiy, Ivan Poliakov, Patrick Olivier, Maisie Rowland, and Emma Foster. Recommender system based on pairwise association rules. *Expert Systems with Applications*, 115:535–542, 2019.

[215] Hyung Jun Ahn. A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem. *Information Sciences*, 178(1):37–51, 2008.

[216] JesúS Bobadilla, Fernando Ortega, Antonio Hernando, and JesúS Bernal. A collaborative filtering approach to mitigate the new user cold start problem. *Knowledge-based systems*, 26:225–238, 2012.

[217] Heung-Nam Kim, Abdulmotaleb El-Saddik, and Geun-Sik Jo. Collaborative error-reflected models for cold-start recommender systems. *Decision Support Systems*, 51(3):519–531, 2011.

[218] Kyung Soo Kim, Doo Soo Chang, and Yong Suk Choi. Boosting memory-based collaborative filtering using content-metadata. *Symmetry*, 11(4):561, 2019.

[219] Pradeep Kumar Singh, Showmik Setta, Pijush Kanti Dutta Pramanik, and Prasenjit Choudhury. Improving the accuracy of collaborative filtering-based recommendations by considering the temporal variance of top-n neighbors. In *International Conference on Innovative Computing and Communications*, pages 1–10. Springer, 2020.

[220] Zhipeng Zhang, Yao Zhang, and Yonggong Ren. Employing neighborhood reduction for alleviating sparsity and cold start problems in user-based collaborative filtering. *Information Retrieval Journal*, 23(4):449–472, 2020.

[221] Chu-Xu Zhang, Zi-Ke Zhang, Lu Yu, Chuang Liu, Hao Liu, and Xiao-Yong Yan. Information filtering via collaborative user clustering modeling. *Physica A: Statistical Mechanics and its Applications*, 396:195–203, 2014.

[222] Manh Cuong Pham, Yiwei Cao, Ralf Klamma, and Matthias Jarke. A clustering approach for collaborative filtering recommendation using social network analysis. *J. UCS*, 17(4):583–604, 2011.

[223] Bin Xu, Jiajun Bu, Chun Chen, and Deng Cai. An exploration of improving collaborative recommender systems via user-item subgroups. In *Proceedings of the 21st international conference on World Wide Web*, pages 21–30. ACM, 2012.

[224] Daqiang Zhang, Ching-Hsien Hsu, Min Chen, Quan Chen, Naixue Xiong, and Jaime Lloret. Cold-start recommendation using bi-clustering and fusion for large-scale social recommender systems. *IEEE Transactions on Emerging Topics in Computing*, 2(2):239–250, 2013.

[225] Xiao Ma, Hongwei Lu, Zaobin Gan, and Qian Zhao. An exploration of improving prediction accuracy by constructing a multi-type clustering based recommendation framework. *Neurocomputing*, 191:388–397, 2016.

[226] Guibing Guo, Jie Zhang, and Neil Yorke-Smith. Leveraging multiviews of trust and similarity to enhance clustering-based recommender systems. *Knowledge-Based Systems*, 74:14–27, 2015.

[227] Andre Luiz Vizine Pereira and Eduardo Raul Hruschka. Simultaneous co-clustering and learning to address the cold start problem in recommender systems. *Knowledge-Based Systems*, 82:11–19, 2015.

[228] Saumya Bansal and Niyati Baliyan. Bi-mars: A bi-clustering based memetic algorithm for recommender systems. *Applied Soft Computing*, 97:106785, 2020.

[229] Xiao Ma, Hongwei Lu, Zaobin Gan, and Jiangfeng Zeng. An explicit trust and distrust clustering based collaborative filtering recommendation approach. *Electronic Commerce Research and Applications*, 25:29–39, 2017.

[230] Jian Liu and Youling Chen. A personalized clustering-based and reliable trust-aware qos prediction approach for cloud service recommendation in cloud manufacturing. *Knowledge-Based Systems*, 174:43–56, 2019.

[231] Zahra Yusefi Hafshejani, Marjan Kaedi, and Afsaneh Fatemi. Improving sparsity and new user problems in collaborative filtering by clustering the personality factors. *Electronic Commerce Research*, 18(4):813–836, 2018.

[232] Mohammed Wasid and Rashid Ali. Fuzzy side information clustering-based framework for effective recommendations. *Computing and Informatics*, 38(3):597–620, 2019.

[233] ThaiBinh Nguyen and Atsuhiro Takasu. A probabilistic model for the cold-start problem in rating prediction using click data. In *International Conference on Neural Information Processing*, pages 196–205. Springer, 2017.

[234] Senthilselvan Natarajan, Subramaniyaswamy Vairavasundaram, Sivaramakrishnan Natarajan, and Amir H Gandomi. Resolving data sparsity and cold start problem in collaborative filtering recommender system using linked open data. *Expert Systems with Applications*, 149:113248, 2020.

[235] Rui Chen, Qingyi Hua, Quanli Gao, and Ying Xing. A hybrid recommender system for gaussian mixture model and enhanced social matrix factorization technology based on multiple interests. *Mathematical Problems in Engineering*, 2018, 2018.

[236] Mehri Davtalab and Ali Asghar Alesheikh. A poi recommendation approach integrating social spatio-temporal information into probabilistic matrix factorization. *Knowledge and Information Systems*, pages 1–21, 2020.

[237] Carl Yang, Lanxiao Bai, Chao Zhang, Quan Yuan, and Jiawei Han. Bridging collaborative filtering and semi-supervised learning: a neural approach for poi recommendation. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1245–1254, 2017.

[238] Jiangbo Shu, Xiaoxuan Shen, Hai Liu, Baolin Yi, and Zhaoli Zhang. A content-based recommendation algorithm for learning resources. *Multimedia Systems*, 24(2):163–173, 2018.

[239] Yong-ping Du, Chang-qing Yao, Shu-hua Huo, and Jing-xuan Liu. A new item-based deep network structure using a restricted boltzmann machine for collaborative filtering. *Frontiers of Information Technology & Electronic Engineering*, 18(5):658–666, 2017.

[240] Ajit P Singh and Geoffrey Gordon. A bayesian matrix factorization model for relational data. *arXiv preprint arXiv:1203.3517*, 2012.

[241] Hao Ma, Dengyong Zhou, Chao Liu, Michael R Lyu, and Irwin King. Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 287–296, 2011.

[242] Zhenghua Xu, Di Yuan, Thomas Lukasiewicz, Cheng Chen, Yishu Miao, and Guizhi Xu. Hybrid deep-semantic matrix factorization for tag-aware personalized recommendation. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3442–3446. IEEE, 2020.

[243] Donghyun Kim, Chanyoung Park, Jinoh Oh, and Hwanjo Yu. Deep hybrid recommender systems via exploiting document context and statistics of items. *Information Sciences*, 417:72–87, 2017.

[244] Donghyuk Shin, Suleyman Cetintas, Kuang-Chih Lee, and Inderjit S Dhillon. Tumblr blog recommendation with boosted inductive matrix completion. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 203–212, 2015.

[245] Travis Ebesu and Yi Fang. Neural semantic personalized ranking for item cold-start recommendation. *Information Retrieval Journal*, 20(2):109–131, 2017.

[246] Yin Zheng, Bangsheng Tang, Wenkui Ding, and Hanning Zhou. A neural autoregressive approach to collaborative filtering. *arXiv preprint arXiv:1605.09477*, 2016.

[247] Vito Bellini, Vito Walter Anelli, Tommaso Di Noia, and Eugenio Di Sciascio. Auto-encoding user ratings via knowledge graphs in recommendation scenarios. In *Proceedings of the 2nd Workshop on Deep Learning for Recommender Systems*, pages 60–66, 2017.

[248] Massimiliano Ruocco, Ole Steinar Lillestøl Skrede, and Helge Langseth. Inter-session modeling for session-based recommendation. In *Proceedings of the 2nd Workshop on Deep Learning for Recommender Systems*, pages 24–31, 2017.

[249] Trinh Xuan Tuan and Tu Minh Phuong. 3d convolutional networks for session-based recommendation with content features. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, pages 138–146, 2017.

[250] Sotirios P Chatzis, Panayiotis Christodoulou, and Andreas S Andreou. Recurrent latent variable networks for session-based recommendation. In *Proceedings of the 2nd Workshop on Deep Learning for Recommender Systems*, pages 38–45, 2017.

[251] Andreu Vall, Hamid Eghbal-Zadeh, Matthias Dorfer, Markus Schedl, and Gerhard Widmer. Music playlist continuation by learning from

hand-curated examples and song features: Alleviating the cold-start problem for rare and out-of-set songs. In *Proceedings of the 2nd Workshop on Deep Learning for Recommender Systems*, pages 46–54, 2017.

[252] Ali Mamdouh Elkahky, Yang Song, and Xiaodong He. A multi-view deep learning approach for cross domain user modeling in recommendation systems. In *Proceedings of the 24th International Conference on World Wide Web*, pages 278–288, 2015.

[253] Gilles Louppe. *Collaborative filtering: Scalable approaches using restricted Boltzmann machines.* PhD thesis, Université de Liège, Liège, Belgique, 2010.

[254] Chao Du, Chongxuan Li, Yin Zheng, Jun Zhu, and Bo Zhang. Collaborative filtering with user-item co-autoregressive models. *arXiv preprint arXiv:1612.07146*, 2016.

[255] Hao Wang, SHI Xingjian, and Dit-Yan Yeung. Collaborative recurrent autoencoder: Recommend while learning to fill in the blanks. In *Advances in Neural Information Processing Systems*, pages 415–423, 2016.

[256] Zhi-Hong Deng, Ling Huang, Chang-Dong Wang, Jian-Huang Lai, and S Yu Philip. Deepcf: A unified framework of representation learning and matching function learning in recommender system. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 61–68, 2019.

[257] Nathan N Liu, Evan W Xiang, Min Zhao, and Qiang Yang. Unifying explicit and implicit feedback for collaborative filtering. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1445–1448, 2010.

[258] Yue Shi, Alexandros Karatzoglou, Linas Baltrunas, Martha Larson, and Alan Hanjalic. xclimf: optimizing expected reciprocal rank for data with multiple levels of relevance. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 431–434, 2013.

[259] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the*

26th International Conference on World Wide Web, pages 173–182. International World Wide Web Conferences Steering Committee, 2017.

[260] Wu-Dong Xi, Ling Huang, Chang-Dong Wang, Yin-Yu Zheng, and Jianhuang Lai. Bpam: Recommendation based on bp neural network with attention mechanism. In *IJCAI*, pages 3905–3911, 2019.

[261] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*, pages 7–10, 2016.

[262] Quangui Zhang, Longbing Cao, Chengzhang Zhu, Zhiqiang Li, and Jinguang Sun. Coupledcf: Learning explicit and implicit user-item couplings in recommendation for deep collaborative filtering. In *IJCAI International Joint Conference on Artificial Intelligence*, 2018.

[263] Yihao Zhang, Chu Zhao, Mian Chen, and Meng Yuan. Integrating stacked sparse auto-encoder into matrix factorization for rating prediction. *IEEE Access*, 9:17641–17648, 2021.

[264] Yi-Cheng Chen, Lin Hui, and Tipajin Thaipisutikul. A collaborative filtering recommendation system with dynamic time decay. *The Journal of Supercomputing*, 77(1):244–262, 2021.

[265] Supriyo Mandal and Abyayananda Maiti. Deep collaborative filtering with social promoter score-based user-item interaction: a new perspective in recommendation. *Applied Intelligence*, pages 1–26, 2021.

[266] M Vijaikumar, Shirish Shevade, and M Narasimha Murty. Gamma: A graph and multi-view memory attention mechanism for top-n heterogeneous recommendation. *Advances in Knowledge Discovery and Data Mining*, 12084:28, 2020.

[267] Huifang Xu, Bo Jiang, and Chris Ding. Mvinf: Social influence prediction with multi-view graph attention learning. *Cognitive Computation*, pages 1–7, 2021.

[268] Daniel Valcarce, Javier Parapar, and Álvaro Barreiro. Finding and analysing good neighbourhoods to improve collaborative filtering. *Knowledge-Based Systems*, 159:193–202, 2018.

[269] Jesús Bobadilla, Antonio Hernando, Fernando Ortega, and Abraham Gutiérrez. Collaborative filtering based on significances. *Information Sciences*, 185(1):1–17, 2012.

[270] Jesús Bobadilla, Francisco Serradilla, and Jesus Bernal. A new collaborative filtering metric that improves the behavior of recommender systems. *Knowledge-Based Systems*, 23(6):520–528, 2010.

[271] Jesús Bobadilla, Fernando Ortega, and Antonio Hernando. A collaborative filtering similarity measure based on singularities. *Information Processing & Management*, 48(2):204–217, 2012.

[272] Jun Wang, Arjen P De Vries, and Marcel JT Reinders. Unified relevance models for rating prediction in collaborative filtering. *ACM Transactions on Information Systems (TOIS)*, 26(3):1–42, 2008.

[273] Aamir Fareed, Saima Hassan, Samir Brahim Belhaouari, and Zahid Halim. A collaborative filtering recommendation framework utilizing social networks. *Machine Learning with Applications*, 14:100495, 2023.

[274] Ninghua Sun, Qiangqiang Luo, Longya Ran, and Peng Jia. Similarity matrix enhanced collaborative filtering for e-government recommendation. *Data & Knowledge Engineering*, 145:102179, 2023.

[275] Shamal Shaikh, Venkateswara Rao Kagita, Vikas Kumar, and Arun K Pujari. Data augmentation and refinement for recommender system: A semi-supervised approach using maximum margin matrix factorization. *Expert Systems with Applications*, 238:121967, 2024.

[276] Krishan Kant Yadav, Hemant Kumar Soni, Ghanshyam Yadav, and Mamta Sharma. Collaborative filtering based hybrid recommendation system using neural network and matrix factorization techniques. *International Journal of Intelligent Systems and Applications in Engineering*, 12(8s):695–701, 2024.

[277] Abinash Pujahari and Dilip Singh Sisodia. Ordinal consistency based matrix factorization model for exploiting side information in collaborative filtering. *Information Sciences*, page 119258, 2023.

[278] Arta Iftikhar, Mustansar Ali Ghazanfar, Mubbashir Ayub, Saad Ali Alahmari, Nadeem Qazi, and Julie Wall. A reinforcement learning recommender system using bi-clustering and markov decision process. *Expert Systems with Applications*, 237:121541, 2024.

[279] Eyad Kannout, Marek Grzegorowski, Michał Grodzki, and Hung Son Nguyen. Clustering-based frequent pattern mining framework for solving cold-start problem in recommender systems. *IEEE Access*, 2024.

[280] Zeinab Shokrzadeh, Mohammad-Reza Feizi-Derakhshi, Mohammad-Ali Balafar, and Jamshid Bagherzadeh Mohasefi. Knowledge graph-based recommendation system enhanced by neural collaborative filtering and knowledge graph embedding. *Ain Shams Engineering Journal*, 15(1):102263, 2024.

[281] Sonu Airen and Jitendra Agrawal. Movie recommender system using parameter tuning of user and movie neighbourhood via co-clustering. *Procedia Computer Science*, 218:1176–1183, 2023.

[282] Jianjun Sun and Qinghua Huang. Two stages biclustering with three populations. *Biomedical Signal Processing and Control*, 79:104182, 2023.

[283] Xin Wang and Serdar Kadıoğlu. Modeling uncertainty to improve personalized recommendations via bayesian deep learning. *International Journal of Data Science and Analytics*, 16(2):191–201, 2023.

[284] Liang-Hong Wu. Bayessentirs: Bayesian sentiment analysis for addressing cold start and sparsity in ranking-based recommender systems. *Expert Systems with Applications*, 238:121930, 2024.

[285] Md Amanatulla, Muppalla Subba Rao, Pothireddy Hemalathareddy, and Kadiyala Pavani. A composite technique for creating contemporary mrs using association rule mining & cf. In *2023 7th International Conference on Computing Methodologies and Communication (ICCMC)*, pages 779–783. IEEE, 2023.

[286] Yuyuan Li, Chaochao Chen, Xiaolin Zheng, Junlin Liu, and Jun Wang. Making recommender systems forget: Learning and unlearning for erasable recommendation. *Knowledge-Based Systems*, 283:111124, 2024.

[287] Mohammed Fadhel Aljunid and Manjaiah Doddaghatta Huchaiah. An efficient hybrid recommendation model based on collaborative filtering recommender systems. *CAAI Transactions on Intelligence Technology*, 6(4):480–492, 2021.

[288] Bogdan Walek and Petr Fajmon. A hybrid recommender system for an online store using a fuzzy expert system. *Expert Systems with Applications*, 212:118565, 2023.

[289] Hamidreza Koohi, Ziad Kobti, Zahra Nazari, and Javad Mousavi. Enhancing recommender system performance through the fusion of fuzzy c-means, restricted boltzmann machine, and extreme learning machine. *Multimedia Tools and Applications*, pages 1–25, 2024.

[290] Monika Verma and Pawan Kumar Patnaik. An automatic college library book recommendation system using optimized hidden markov based weighted fuzzy ranking model. *Engineering Applications of Artificial Intelligence*, 130:107664, 2024.

[291] Yan Zhu. Personalized recommendation of educational resource information based on adaptive genetic algorithm. *International Journal of Reliability, Quality and Safety Engineering*, 30(02):2250014, 2023.

[292] R Kirubahari and S Miruna Joe Amali. An improved restricted boltzmann machine using bayesian optimization for recommender systems. *Evolving Systems*, pages 1–13, 2023.

[293] Diana Ferreira, Sofia Silva, António Abelha, and José Machado. Recommendation system using autoencoders. *Applied Sciences*, 10(16):5510, 2020.

[294] Rui Hou. Music content personalized recommendation system based on a convolutional neural network. *Soft Computing*, 28(2):1785–1802, 2024.

[295] Shivangi Gheewala, Shuxiang Xu, Soonja Yeom, and Sumbal Maqsood. Exploiting deep transformer models in textual review based recommender systems. *Expert Systems with Applications*, 235:121120, 2024.

[296] Chao Yang, Lianhai Miao, Bin Jiang, Dongsheng Li, and Da Cao. Gated and attentive neural collaborative filtering for user generated list recommendation. *Knowledge-based systems*, 187:104839, 2020.

[297] Hao Wu, Zhengxin Zhang, Kun Yue, Binbin Zhang, and Ruichao Zhu. Content embedding regularized matrix factorization for recommender systems. In *2017 IEEE International Congress on Big Data (BigData Congress)*, pages 209–215. IEEE, 2017.

[298] Ruiqin Wang, Zongda Wu, Jungang Lou, and Yunliang Jiang. Attention-based dynamic user modeling and deep collaborative filtering recommendation. *Expert Systems with Applications*, 188:116036, 2022.

[299] Jinjin Zhang, Chenhui Ma, Xiaodong Mu, Peng Zhao, Chengliang Zhong, and A Ruhan. Recurrent convolutional neural network for session-based recommendation. *Neurocomputing*, 437:157–167, 2021.

[300] Daniel Valcarce, Alfonso Landin, Javier Parapar, and Álvaro Barreiro. Collaborative filtering embeddings for memory-based recommender systems. *Engineering Applications of Artificial Intelligence*, 85:347–356, 2019.

[301] Qi Liu, Enhong Chen, Hui Xiong, Chris HQ Ding, and Jian Chen. Enhancing collaborative filtering by user interest expansion via personalized ranking. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(1):218–233, 2011.

[302] Gene H Golub and Christian Reinsch. Singular value decomposition and least squares solutions. In *Handbook for Automatic Computation: Volume II: Linear Algebra*, pages 134–151. Springer, 1971.

[303] Hung-Hsu Chou, Carsten Gieshoff, Johannes Maly, and Holger Rauhut. Gradient descent for deep matrix factorization: Dynamics and implicit bias towards low rank. *Applied and Computational Harmonic Analysis*, 68:101595, 2024.

[304] Reetu Singh, Pragya Dwivedi, and Pankaj Patidar. Multi-criteria recommendation system based on deep matrix factorization and regression techniques. *International Journal of Information Technology*, pages 1–12, 2024.

[305] Farid Saberi-Movahed, Bitasta Biswas, Prayag Tiwari, Jens Lehmann, and Sahar Vahdati. Deep nonnegative matrix factorization with joint global and local structure preservation. *Expert Systems with Applications*, 249:123645, 2024.

[306] Jiewen Guan, Bilian Chen, and Shenbao Yu. A hybrid similarity model for mitigating the cold-start problem of collaborative filtering in sparse data. *Expert Systems with Applications*, 249:123700, 2024.

[307] Dongeon Kim, Qinglong Li, Dongsoo Jang, and Jaekyeong Kim. Axcf: Aspect-based collaborative filtering for explainable recommendations. *Expert Systems*, page e13594.

[308] Mengxing Huang, Xiu Shi Zhang, Uzair Aslam Bhatti, YuanYuan Wu, Yu Zhang, and Yazeed Yasin Ghadi. An interpretable approach using hybrid graph networks and explainable ai for intelligent diagnosis recommendations in chronic disease care. *Biomedical Signal Processing and Control*, 91:105913, 2024.

[309] Mikhail Genkin and JJ McArthur. A transfer learning approach to minimize reinforcement learning risks in energy optimization for automated and smart buildings. *Energy and Buildings*, 303:113760, 2024.

[310] Meenakshi Aggarwal, Vikas Khullar, Nitin Goyal, Rashi Rastogi, Aman Singh, Vanessa Yelamos Torres, and Marwan Ali Albahar. Privacy preserved collaborative transfer learning model with heterogeneous distributed data for brain tumor classification. *International Journal of Imaging Systems and Technology*, 34(2):e22994, 2024.

[311] Xiaokang Zhou, Qiuyue Yang, Qiang Liu, Wei Liang, Kevin Wang, Zhi Liu, Jianhua Ma, and Qun Jin. Spatial–temporal federated transfer learning with multi-sensor data fusion for cooperative positioning. *Information Fusion*, 105:102182, 2024.

[312] Stelios Giannikis, Flavius Frasincar, and David Boekestijn. Reinforcement learning for addressing the cold-user problem in recommender systems. *Knowledge-Based Systems*, page 111752, 2024.

[313] Aobo Xu, Ling Jian, Yue Yin, and Na Zhang. Uisa: User information separating architecture for commodity recommendation policy with deep reinforcement learning. *ACM Transactions on Recommender Systems*, 2024.

[314] Muhammad Alrashidi, Ali Selamat, Roliana Ibrahim, and Hamido Fujita. Social recommender system based on cnn incorporating tagging and contextual features. *Journal of Cases on Information Technology (JCIT)*, 26(1):1–20, 2024.

[315] Yashar Deldjoo, Dietmar Jannach, Alejandro Bellogin, Alessandro Difonzo, and Dario Zanzonelli. Fairness in recommender systems: research landscape and future directions. *User Modeling and User-Adapted Interaction*, 34(1):59–108, 2024.

[316] Yashar Deldjoo. Understanding biases in chatgpt-based recommender systems: Provider fairness, temporal stability, and recency. *arXiv preprint arXiv:2401.10545*, 2024.

[317] Talha Iqbal, Mehedi Masud, Bilal Amin, Conor Feely, Mary Faherty, Tim Jones, Michelle Tierney, Atif Shahzad, and Patricia Vazquez. Towards integration of artificial intelligence into medical devices as a real-time recommender system for personalised healthcare: State-of-the-art and future prospects. *Health Sciences Review*, page 100150, 2024.

[318] Alexey Mikhaylov, Ishaq M Bhatti, Hasan Dinçer, and Serhat Yüksel. Integrated decision recommendation system using iteration-enhanced collaborative filtering, golden cut bipolar for analyzing the risk-based oil market spillovers. *Computational Economics*, 63(1):305–338, 2024.

[319] Yiwen Zhang, Vladimir Y Mariano, and Rex P Bringula. Prediction of students' grade by combining educational knowledge graph and collaborative filtering. *IEEE Access*, 2024.

[320] Haithem Mezni, Mokhtar Sellami, Amal Al-Rasheed, and Hela Elmannai. Cross-network service recommendation in smart cities. *Concurrency and Computation: Practice and Experience*, page e8063, 2024.

Dr. Mohammed Fadhel Aljunid is a data scientist at Thamar Center of Computer & Informatics, Faculty of Computers & Informatics, Thamar University and director of Information Systems Unit of the Investment Sector at the Higher Agricultural and Fisheries Committee in Yemen. He holds a Doctor of Philosophy in Computer Science (Ph.D.) from Mangalore University, India and specializes on recommendation systems, natural language processing, sentiment analysis using machine learning and deep learning. He has published several articles in leading peer-reviewed journals such as Journal of Expert System with Application and CAAI Transactions on Intelligence Technology. He obtained M.Sc. from Mysore University, India and a Bachelor of Science in Computer Science (B.Sc.) from Thamar University, Yemen.

Dr. Manjaiah D.H. is a Senior Professor and Chairman in the Department of PG Studies and Research in Computer Science at Mangalore University. He holds a PhD in Advanced Network Computing Paradigm with Quality of Services and Performance Analysis. His expertise includes Routing Algorithms for Mobile Ad-Hoc networks, Big data analytics, Recommendation system, Vertical Handoff Mechanisms for 4G Networks, and Digital Forensic Laboratory enhancement in India. He has led numerous research projects sponsored by UGC, DST-SERB, and DST-VGST. He has authored and published around 153 in reputed International / National journals and conferences on various topics in Mobile AD -

HOC Networks, IPv4 and IPv6 mobility, Machine Learning and Deep Learning, Virtualization using Cloud Computing / Cyber security and Digital forensics.

Dr. Mohammad Kazim Hooshmand is a data scientist and university lecturer at Kabul Education University in the Department of Computer Science. He holds a Ph.D in Computer Science from Mangalore University and specializes in network anomaly detection modeling and data analysis. Dr. Mohammad Kazim Hooshmand research focuses on anomaly detection in network traffic data using machine learning and deep learning techniques. He has published several articles in leading peer-reviewed journals such as Journal of CAAI Transactions on Intelligence Technology.

Mr. Sadiq Qaid Alzoubah is a researcher in the Department of Computer Science at Mangalore University, India in the area of recommender system. He has obtained an MSc from the Department of Computer Science and has Department of Computer Science Mangalore University for PG students and a Bachelor of Science in Computer Science (B.Sc.) from Thamar University, Yemen.

Ms. Amrithkala M Shetty is a researcher in the Department of Computer Science. She has obtained an MSc from the Department of Computer Science and has over 10 years of teaching experience at Mangalore University for PG students. Her expertise includes machine learning, deep learning, cybersecurity, and sentiment analysis.

Mr. Wasim A. Ali received his B.E. in Information Technology from University of Science and Technology, Taiz, Yemen (2014). He earned his master's degree in computer science from St. Philomena's College, University of Mysore, India(2017). PhD in Electrical and Information Engineering at Department of Electrical and Information Engineering, Politecnico di Bari, Italy (2020-2024). Currently, work as a research assistant in Integrating Network Digital Twinning into Future AI-based 6G Systems - Horizon Europe Framework Programme (HORIZON) at LCA - Laboratory of Control and Automation, Polytechnic University of Bari

Declaration of interests

☐The authors declare that they have no known competing financialinterestsor personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

☐The authors declare that they have no known competing financialinterestsor personal relationships that could have appeared to influence the work reported in this paper.