https://doi.org/10.1186/s40537-022-00592-5

Journal of Big Data

SURVEY PAPER Open Access

A systematic review and research perspective on recommender systems

Deepjyoti Roy[*] and Mala Dutta

*Correspondence:
deepjyoti2roy@gmail.com
Department of Computer Science & Engineering, Assam Down Town University, Panikhaiti, Guwahati 781026, Assam, India

Abstract

Recommender systems are efficient tools for filtering online information, which is widespread owing to the changing habits of computer users, personalization trends, and emerging access to the internet. Even though the recent recommender systems are eminent in giving precise recommendations, they suffer from various limitations and challenges like scalability, cold-start, sparsity, etc. Due to the existence of various techniques, the selection of techniques becomes a complex work while building application-focused recommender systems. In addition, each technique comes with its own set of features, advantages and disadvantages which raises even more questions, which should be addressed. This paper aims to undergo a systematic review on various recent contributions in the domain of recommender systems, focusing on diverse applications like books, movies, products, etc. Initially, the various applications of each recommender system are analysed. Then, the algorithmic analysis on various recommender systems is performed and a taxonomy is framed that accounts for various components required for developing an effective recommender system. In addition, the datasets gathered, simulation platform, and performance metrics focused on each contribution are evaluated and noted. Finally, this review provides a much-needed overview of the current state of research in this field and points out the existing gaps and challenges to help posterity in developing an efficient recommender system.

Keywords: Recommender system, Machine learning, Content-based filtering, Collaborative filtering, Deep learning, Review

Introduction

The recent advancements in technology along with the prevalence of online services has offered more abilities for accessing a huge amount of online information in a faster manner. Users can post reviews, comments, and ratings for various types of services and products available online. However, the recent advancements in pervasive computing have resulted in an online data overload problem. This data overload complicates the process of finding relevant and useful content over the internet. The recent establishment of several procedures having lower computational requirements can however guide users to the relevant content in a much easy and fast manner. Because of this, the development of recommender systems has recently gained significant attention. In general, recommender systems act as information filtering tools, offering users suitable and

Springer Open

personalized content or information. Recommender systems primarily aim to reduce the user's effort and time required for searching relevant information over the internet.

Nowadays, recommender systems are being increasingly used for a large number of applications such as web [1, 67, 70], books [2], e-learning [4, 16, 61], tourism [5, 8, 78], movies [66], music [79], e-commerce, news, specialized research resources [65], television programs [72, 81], etc. It is therefore important to build high-quality and exclusive recommender systems for providing personalized recommendations to the users in various applications. Despite the various advances in recommender systems, the present generation of recommender systems requires further improvements to provide more efficient recommendations applicable to a broader range of applications. More investigation of the existing latest works on recommender systems is required which focus on diverse applications.

There is hardly any review paper that has categorically synthesized and reviewed the literature of all the classification fields and application domains of recommender systems. The few existing literature reviews in the field cover just a fraction of the articles or focus only on selected aspects such as system evaluation. Thus, they do not provide an overview of the application field, algorithmic categorization, or identify the most promising approaches. Also, review papers often neglect to analyze the dataset description and the simulation platforms used. This paper aims to fulfil this significant gap by reviewing and comparing existing articles on recommender systems based on a defined classification framework, their algorithmic categorization, simulation platforms used, applications focused, their features and challenges, dataset description and system performance. Finally, we provide researchers and practitioners with insight into the most promising directions for further investigation in the field of recommender systems under various applications.

In essence, recommender systems deal with two entities—users and items, where each user gives a rating (or preference value) to an item (or product). User ratings are generally collected by using implicit or explicit methods. Implicit ratings are collected indirectly from the user through the user's interaction with the items. Explicit ratings, on the other hand, are given directly by the user by picking a value on some finite scale of points or labelled interval values. For example, a website may obtain implicit ratings for different items based on clickstream data or from the amount of time a user spends on a webpage and so on. Most recommender systems gather user ratings through both explicit and implicit methods. These feedbacks or ratings provided by the user are arranged in a user-item matrix called the utility matrix as presented in Table 1.

The utility matrix often contains many missing values. The problem of recommender systems is mainly focused on finding the values which are missing in the utility matrix.

Table 1 Utility matrix

	Item 1	**Item 2**	**Item 3**	**Item 4**
User 1	3	–	2	–
User 2	–	4	–	2
User 3	4	–	3	–
User 4	–	5	4	1

This task is often difficult as the initial matrix is usually very sparse because users generally tend to rate only a small number of items. It may also be noted that we are interested in only the high user ratings because only such items would be suggested back to the users. The efficiency of a recommender system greatly depends on the type of algorithm used and the nature of the data source—which may be contextual, textual, visual etc.

Types of recommender systems

Recommender systems are broadly categorized into three different types viz. content-based recommender systems, collaborative recommender systems and hybrid recommender systems. A diagrammatic representation of the different types of recommender systems is given in Fig. 1.

Content-based recommender system

In content-based recommender systems, all the data items are collected into different item profiles based on their description or features. For example, in the case of a book, the features will be author, publisher, etc. In the case of a movie, the features will be the movie director, actor, etc. When a user gives a positive rating to an item, then the other items present in that item profile are aggregated together to build a user profile. This user profile combines all the item profiles, whose items are rated positively by the user. Items present in this user profile are then recommended to the user, as shown in Fig. 2.

One drawback of this approach is that it demands in-depth knowledge of the item features for an accurate recommendation. This knowledge or information may not be always available for all items. Also, this approach has limited capacity to expand on the users' existing choices or interests. However, this approach has many advantages. As user preferences tend to change with time, this approach has the quick capability of dynamically adapting itself to the changing user preferences. Since one user profile is specific only to that user, this algorithm does not require the profile details of any other users because they provide no influence in the recommendation process. This ensures the security and privacy of user data. If new items have sufficient description, content-based techniques can overcome the cold-start problem i.e., this technique can recommend an

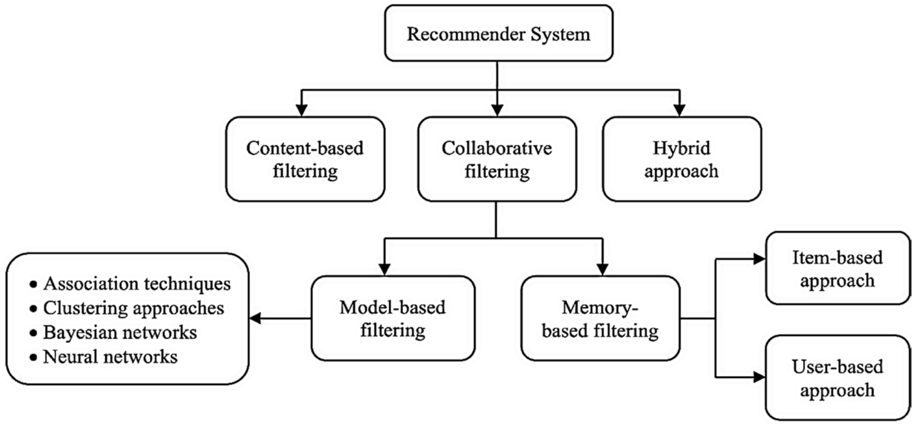

Fig. 1 Types of recommender systems

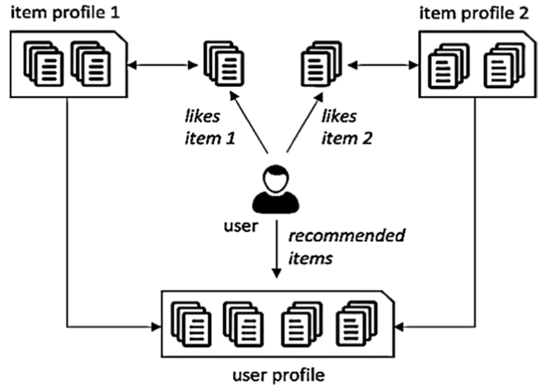

Fig. 2 Content-based recommender system

item even when that item has not been previously rated by any user. Content-based filtering approaches are more common in systems like personalized news recommender systems, publications, web pages recommender systems, etc.

Collaborative filtering-based recommender system

Collaborative approaches make use of the measure of similarity between users. This technique starts with finding a group or collection of user X whose preferences, likes, and dislikes are similar to that of user A. X is called the neighbourhood of A. The new items which are liked by most of the users in X are then recommended to user A. The efficiency of a collaborative algorithm depends on how accurately the algorithm can find the neighbourhood of the target user. Traditionally collaborative filtering-based systems suffer from the cold-start problem and privacy concerns as there is a need to share user data. However, collaborative filtering approaches do not require any knowledge of item features for generating a recommendation. Also, this approach can help to expand on the user's existing interests by discovering new items. Collaborative approaches are again divided into two types: memory-based approaches and model-based approaches.

Memory-based collaborative approaches recommend new items by taking into consideration the preferences of its neighbourhood. They make use of the utility matrix directly for prediction. In this approach, the first step is to build a model. The model is equal to a function that takes the utility matrix as input.

$$Model = f\,(utility\ matrix)$$

Then recommendations are made based on a function that takes the model and user profile as input. Here we can make recommendations only to users whose user profile belongs to the utility matrix. Therefore, to make recommendations for a new user, the user profile must be added to the utility matrix, and the similarity matrix should be recomputed, which makes this technique computation heavy.

$$Recommendation = f\,(defined\ model,\ user\ profile)$$
$$where\ user\ profile \in utility\ matrix$$

Memory-based collaborative approaches are again sub-divided into two types: user-based collaborative filtering and item-based collaborative filtering. In the user-based

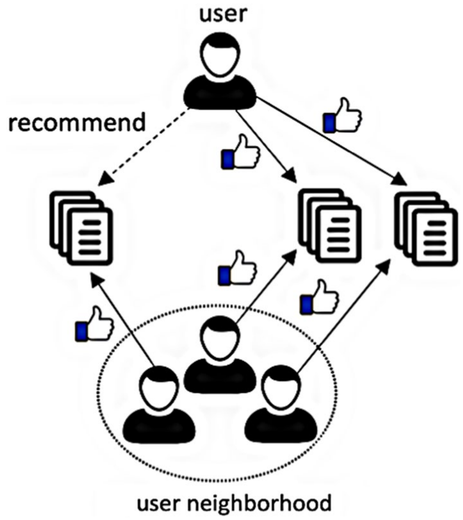

Fig. 3 User-based collaborative filtering

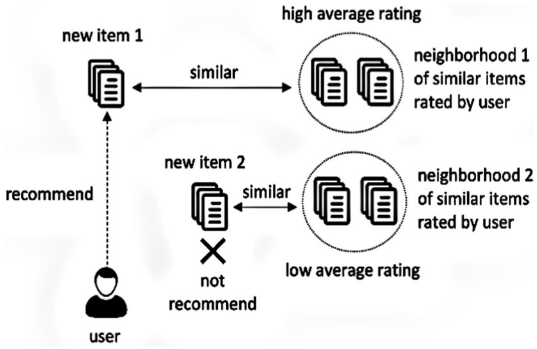

Fig. 4 Item-based collaborative filtering

approach, the user rating of a new item is calculated by finding other users from the user neighbourhood who has previously rated that same item. If a new item receives positive ratings from the user neighbourhood, the new item is recommended to the user. Figure 3 depicts the user-based filtering approach.

In the item-based approach, an item-neighbourhood is built consisting of all similar items which the user has rated previously. Then that user's rating for a different new item is predicted by calculating the weighted average of all ratings present in a similar item-neighbourhood as shown in Fig. 4.

Model-based systems use various data mining and machine learning algorithms to develop a model for predicting the user's rating for an unrated item. They do not rely on the complete dataset when recommendations are computed but extract features from the dataset to compute a model. Hence the name, model-based technique. These techniques also need two steps for prediction—the first step is to build the model, and the second step is to predict ratings using a function (f) which takes the model defined in the first step and the user profile as input.

$$\begin{aligned}&Recommendation = f\,(defined\ model,\ user\ profile)\\&where\ user\ profile \not\subset utility\ matrix\end{aligned}$$

Model-based techniques do not require adding the user profile of a new user into the utility matrix before making predictions. We can make recommendations even to users that are not present in the model. Model-based systems are more efficient for group recommendations. They can quickly recommend a group of items by using the pre-trained model. The accuracy of this technique largely relies on the efficiency of the underlying learning algorithm used to create the model. Model-based techniques are capable of solving some traditional problems of recommender systems such as sparsity and scalability by employing dimensionality reduction techniques [86] and model learning techniques.

Hybrid filtering

A hybrid technique is an aggregation of two or more techniques employed together for addressing the limitations of individual recommender techniques. The incorporation of different techniques can be performed in various ways. A hybrid algorithm may incorporate the results achieved from separate techniques, or it can use content-based filtering in a collaborative method or use a collaborative filtering technique in a content-based method. This hybrid incorporation of different techniques generally results in increased performance and increased accuracy in many recommender applications. Some of the hybridization approaches are meta-level, feature-augmentation, feature-combination, mixed hybridization, cascade hybridization, switching hybridization and weighted hybridization [86]. Table 2 describes these approaches.

Recommender system challenges

This section briefly describes the various challenges present in current recommender systems and offers different solutions to overcome these challenges.

Cold start problem

The cold start problem appears when the recommender system cannot draw any inference from the existing data, which is insufficient. Cold start refers to a condition when the system cannot produce efficient recommendations for the cold (or new) users who have not rated any item or have rated a very few items. It generally arises when a new user enters the system or new items (or products) are inserted into the database.

Table 2 Hybrid filtering methods

Hybrid methods	Description
Meta-level	A pre-learned model is used as an input to another recommender system
Feature combination	The features of one recommender system are injected into another
Feature augmentation	The result of one model is applied as an input to another
Mixed hybridization	The output of different recommender systems are mixed, and the combined result is given as a recommendation
Cascade hybridization	One system improvises the output of another
Switching hybridization	Select one recommender model based on the current requirement
Weighted hybridization	Ratings of different techniques are aggregated to compute a single recommendation

Some solutions to this problem are as follows: (a) Ask new users to explicitly mention their item preference. (b) Ask a new user to rate some items at the beginning. (c) Collect demographic information (or meta-data) from the user and recommend items accordingly.

Shilling attack problem

This problem arises when a malicious user fakes his identity and enters the system to give false item ratings [87]. Such a situation occurs when the malicious user wants to either increase or decrease some item's popularity by causing a bias on selected target items. Shilling attacks greatly reduce the reliability of the system. One solution to this problem is to detect the attackers quickly and remove the fake ratings and fake user profiles from the system.

Synonymy problem

This problem arises when similar or related items have different entries or names, or when the same item is represented by two or more names in the system [78]. For example, babywear and baby cloth. Many recommender systems fail to distinguish these differences, hence reducing their recommendation accuracy. To alleviate this problem many methods are used such as demographic filtering, automatic term expansion and Singular Value Decomposition [76].

Latency problem

The latency problem is specific to collaborative filtering approaches and occurs when new items are frequently inserted into the database. This problem is characterized by the system's failure to recommend new items. This happens because new items must be reviewed before they can be recommended in a collaborative filtering environment. Using content-based filtering may resolve this issue, but it may introduce overspecialization and decrease the computing time and system performance. To increase performance, the calculations can be done in an offline environment and clustering-based techniques can be used [76].

Sparsity problem

Data sparsity is a common problem in large scale data analysis, which arises when certain expected values are missing in the dataset. In the case of recommender systems, this situation occurs when the active users rate very few items. This reduces the recommendation accuracy. To alleviate this problem several techniques can be used such as demographic filtering, singular value decomposition and using model-based collaborative techniques.

Grey sheep problem

The grey sheep problem is specific to pure collaborative filtering approaches where the feedback given by one user do not match any user neighbourhood. In this situation, the system fails to accurately predict relevant items for that user. This problem can be resolved by using pure content-based approaches where predictions are made based on the user's profile and item properties.

Scalability problem

Recommender systems, especially those employing collaborative filtering techniques, require large amounts of training data, which cause scalability problems. The scalability problem arises when the amount of data used as input to a recommender system increases quickly. In this era of big data, more and more items and users are rapidly getting added to the system and this problem is becoming common in recommender systems. Two common approaches used to solve the scalability problem is dimensionality reduction and using clustering-based techniques to find users in tiny clusters instead of the complete database.

Methodology

The purpose of this study is to understand the research trends in the field of recommender systems. The nature of research in recommender systems is such that it is difficult to confine each paper to a specific discipline. This can be further understood by the fact that research papers on recommender systems are scattered across various journals such as computer science, management, marketing, information technology and information science. Hence, this literature review is conducted over a wide range of electronic journals and research databases such as ACM Portal, IEEE/IEE Library, Google Scholars and Science Direct [88].

The search process of online research articles was performed based on 6 descriptors: "Recommender systems", "Recommendation systems", "Movie Recommend*", "Music Recommend*", "Personalized Recommend*", "Hybrid Recommend*". The following research papers described below were excluded from our research:

- News articles.
- Master's dissertations.
- Non-English papers.
- Unpublished papers.
- Research papers published before 2011.

We have screened a total of 350 articles based on their abstracts and content. However, only research papers that described how recommender systems can be applied were chosen. Finally, 60 papers were selected from top international journals indexed in Scopus or E-SCI in 2021. We now present the PRISMA flowchart of the inclusion and exclusion process in Fig. 5.

Each paper was carefully reviewed and classified into 6 categories in the application fields and 3 categories in the techniques used to develop the system. The classification framework is presented in Fig. 6.

The number of relevant articles come from Expert Systems with Applications (23%), followed by IEEE (17%), Knowledge-Based System (17%) and Others (43%). Table 3 depicts the article distribution by journal title and Table 4 depicts the sector-wise article distribution.

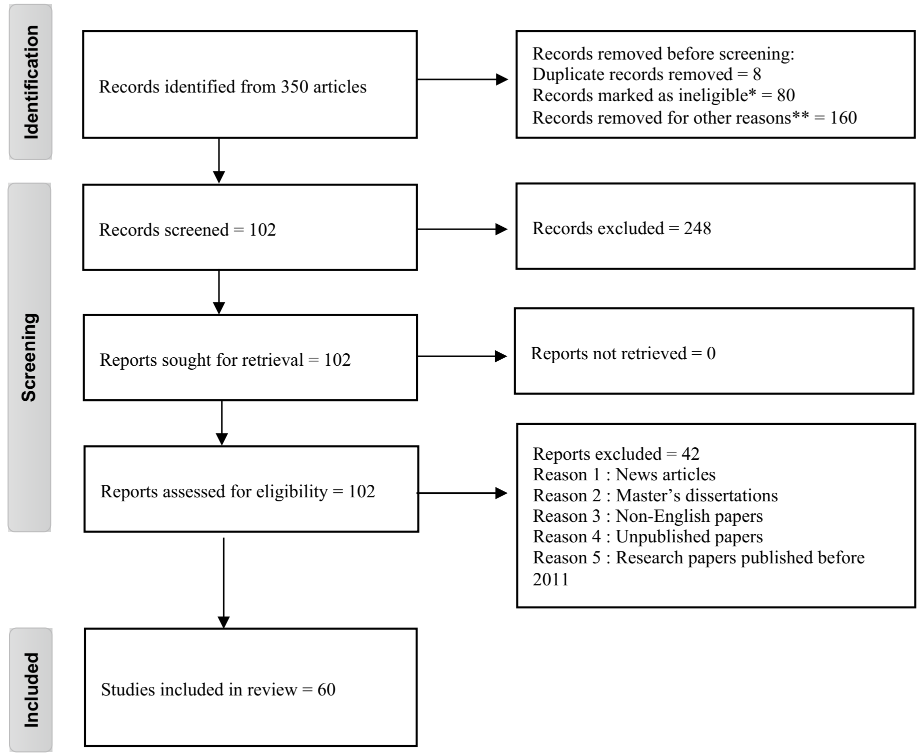
Identification
Records identified from 350 articles
Records removed before screening:
Duplicate records removed = 8
Records marked as ineligible* = 80
Records removed for other reasons** = 160
Screening
Records screened = 102
Records excluded = 248
Reports sought for retrieval = 102
Reports not retrieved = 0
Reports assessed for eligibility = 102
Reports excluded = 42
Reason 1 : News articles
Reason 2 : Master's dissertations
Reason 3 : Non-English papers
Reason 4 : Unpublished papers
Reason 5 : Research papers published before 2011
Included
Studies included in review = 60

Fig. 5 PRISMA flowchart of the inclusion and exclusion process. Abstract and content not suitable to the study: *The use or application of the recommender system is not specified: **

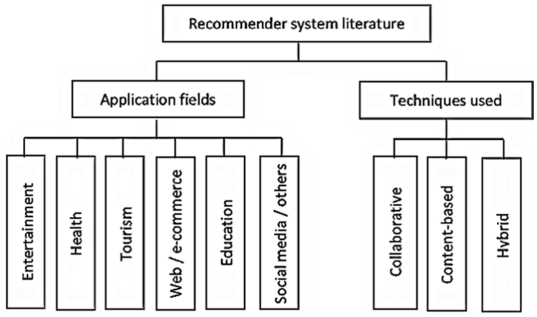

Fig. 6 Classification framework

Table 3 Article distribution by journal title

Journal title	No	% (approx.)
Expert Systems with Applications	14	23
IEEE Access	10	17
Knowledge-Based Systems	10	17
Others (various Elsevier journals)	26	43
Total	60	100

Table 4 Sector-wise article distribution

Sector	No	% (approx.)
Health	8	13
Entertainment	16	26
Tourism	7	12
Web/e-commerce	10	17
Education	7	12
Social media/others	12	20
Total	60	100

Both forward and backward searching techniques were implemented to establish that the review of 60 chosen articles can represent the domain literature. Hence, this paper can demonstrate its validity and reliability as a literature review.

Review on state-of-the-art recommender systems

This section presents a state-of-art literature review followed by a chronological review of the various existing recommender systems.

Literature review

In 2011, Castellano et al. [1] developed a "NEuro-fuzzy WEb Recommendation (NEWER)" system for exploiting the possibility of combining computational intelligence and user preference for suggesting interesting web pages to the user in a dynamic environment. It considered a set of fuzzy rules to express the correlations between user relevance and categories of pages. Crespo et al. [2] presented a recommender system for distance education over internet. It aims to recommend e-books to students using data from user interaction. The system was developed using a collaborative approach and focused on solving the data overload problem in big digital content. Lin et al. [3] have put forward a recommender system for automatic vending machines using Genetic algorithm (GA), k-means, Decision Tree (DT) and Bayesian Network (BN). It aimed at recommending localized products by developing a hybrid model combining statistical methods, classification methods, clustering methods, and meta-heuristic methods. Wang and Wu [4] have implemented a ubiquitous learning system for providing personalized learning assistance to the learners by combining the recommendation algorithm with a context-aware technique. It employed the Association Rule Mining (ARM) technique and aimed to increase the effectiveness of the learner's learning. García-Crespo et al. [5] presented a "semantic hotel" recommender system by considering the experiences of consumers using a fuzzy logic approach. The system considered both hotel and customer characteristics. Dong et al. [6] proposed a structure for a service-concept recommender system using a semantic similarity model by integrating the techniques from the view of an ontology structure-oriented metric and a concept content-oriented metric. The system was able to deliver optimal performance when compared with similar recommender systems. Li et al. [7] developed a Fuzzy linguistic modelling-based recommender system for assisting users to find experts in knowledge management systems. The developed system was applied to the aircraft industry where it demonstrated efficient and feasible performance. Lorenzi et al. [8] presented an "assumption-based multiagent" system to make travel package recommendations using user preferences in the tourism industry. It performed different tasks like discovering, filtering, and integrating specific information for building a travel package following the user requirement. Huang et al. [9] proposed a context-aware recommender system through the extraction, evaluation and incorporation of contextual information gathered using the collaborative filtering and rough set model.

In 2012, Chen et al. [10] presented a diabetes medication recommender model by using "Semantic Web Rule Language (SWRL) and Java Expert System Shell (JESS)" for aggregating suitable prescriptions for the patients. It aimed at selecting the most suitable drugs from the list of specific drugs. Mohanraj et al. [11] developed the "Ontology-driven bee's foraging approach (ODBFA)" to accurately predict the online navigations most likely to be visited by a user. The self-adaptive system is intended to capture the various requirements of the online user by using a scoring technique and by performing a similarity comparison. Hsu et al. [12] proposed a "personalized auxiliary material" recommender system by considering the specific course topics, individual learning styles, complexity of the auxiliary materials using an artificial bee colony algorithm. Gemmell et al. [13] demonstrated a solution for the problem of resource recommendation in social annotation systems. The model was developed using a linear-weighted hybrid method which was capable of providing recommendations under different constraints. Choi et al. [14] proposed one "Hybrid Online-Product rEcommendation (HOPE) system" by the integration of collaborative filtering through sequential pattern analysis-based recommendations and implicit ratings. Garibaldi et al. [15] put forward a technique for incorporating the variability in a fuzzy inference model by using non-stationary fuzzy sets for replicating the variabilities of a human. This model was applied to a decision problem for treatment recommendations of post-operative breast cancer.

In 2013, Salehi and Kmalabadi [16] proposed an e-learning material recommender system by "modelling of materials in a multidimensional space of material's attribute". It employed both content and collaborative filtering. Aher and Lobo [17] introduced a course recommender system using data mining techniques such as simple K-means clustering and Association Rule Mining (ARM) algorithm. The proposed e-learning system was successfully demonstrated for "MOOC (Massively Open Online Courses)". Kardan and Ebrahimi [18] developed a hybrid recommender system for recommending posts in asynchronous discussion groups. The system was built combining both collaborative filtering and content-based filtering. It considered implicit user data to compute the user similarity with various groups, for recommending suitable posts and contents to its users. Chang et al. [19] adopted a cloud computing technology for building a TV program recommender system. The system designed for digital TV programs was implemented using Hadoop Fair Scheduler (HFC), K-means clustering and k-nearest neighbour (KNN) algorithms. It was successful in processing huge amounts of real-time user data. Lucas et al. [20] implemented a recommender model for assisting a tourism application by using associative classification and fuzzy logic to predict the context. Niu et al. [21] introduced "Affivir: An Affect-based Internet

Video Recommendation System" which was developed by calculating user preferences and by using spectral clustering. This model recommended videos with similar effects, which was processed to get optimal results with dynamic adjustments of recommendation constraints.

In 2014, Liu et al. [22] implemented a new route recommendation model for offering personalized and real-time route recommendations for self-driven tourists to minimize the queuing time and traffic jams infamous tourist places. Recommendations were carried out by considering the preferences of users. Bakshi et al. [23] proposed an unsupervised learning-based recommender model for solving the scalability problem of recommender systems. The algorithm used transitive similarities along with Particle Swarm Optimization (PSO) technique for discovering the global neighbours. Kim and Shim [24] proposed a recommender system based on "latent Dirichlet allocation using probabilistic modelling for Twitter" that could recommend the top-K tweets for a user to read, and the top-K users to follow. The model parameters were learned from an inference technique by using the differential Expectation–Maximization (EM) algorithm. Wang et al. [25] developed a hybrid-movie recommender model by aggregating a genetic algorithm (GA) with improved K-means and Principal Component Analysis (PCA) technique. It was able to offer intelligent movie recommendations with personalized suggestions. Kolomvatsos et al. [26] proposed a recommender system by considering an optimal stopping theory for delivering books or music recommendations to the users. Gottschlich et al. [27] proposed a decision support system for stock investment recommendations. It computed the output by considering the overall crowd's recommendations. Torshizi et al. [28] have introduced a hybrid recommender system to determine the severity level of a medical condition. It could recommend suitable therapies for patients suffering from Benign Prostatic Hyperplasia.

In 2015, Zahálka et al. [29] proposed a venue recommender: "City Melange". It was an interactive content-based model which used the convolutional deep-net features of the visual domain and the linear Support Vector Machine (SVM) model to capture the semantic information and extract latent topics. Sankar et al. [30] have proposed a stock recommender system based on the stock holding portfolio of trusted mutual funds. The system employed the collaborative filtering approach along with social network analysis for offering a decision support system to build a trust-based recommendation model. Chen et al. [31] have put forward a novel movie recommender system by applying the "artificial immune network to collaborative filtering" technique. It computed the affinity of an antigen and the affinity between an antibody and antigen. Based on this computation a similarity estimation formula was introduced which was used for the movie recommendation process. Wu et al. [32] have examined the technique of data fusion for increasing the efficiency of item recommender systems. It employed a hybrid linear combination model and used a collaborative tagging system. Yeh and Cheng [33] have proposed a recommender system for tourist attractions by constructing the "elicitation mechanism using the Delphi panel method and matrix construction mechanism using the repertory grids", which was developed by considering the user preference and expert knowledge.

In 2016, Liao et al. [34] proposed a recommender model for online customers using a rough set association rule. The model computed the probable behavioural variations of online consumers and provided product category recommendations for e-commerce platforms. Li et al. [35] have suggested a movie recommender system based on user feedback collected from microblogs and social networks. It employed the sentiment-aware association rule mining algorithm for recommendations using the prior information of frequent program patterns, program metadata similarity and program view logs. Wu et al. [36] have developed a recommender system for social media platforms by aggregating the technique of Social Matrix Factorization (SMF) and Collaborative Topic Regression (CTR). The model was able to compute the ratings of users to items for making recommendations. For improving the recommendation quality, it gathered information from multiple sources such as item properties, social networks, feedback, etc. Adeniyi et al. [37] put forward a study of automated web-usage data mining and developed a recommender system that was tested in both real-time and online for identifying the visitor's or client's clickstream data.

In 2017, Rawat and Kankanhalli [38] have proposed a viewpoint recommender system called "ClickSmart" for assisting mobile users to capture high-quality photographs at famous tourist places. Yang et al. [39] proposed a gradient boosting-based job recommendation system for satisfying the cost-sensitive requirements of the users. The hybrid algorithm aimed to reduce the rate of unnecessary job recommendations. Lee et al. [40] proposed a music streaming recommender system based on smartphone activity usage. The proposed system benefitted by using feature selection approaches with machine learning techniques such as Naive Bayes (NB), Support Vector Machine (SVM), Multi-layer Perception (MLP), Instance-based k -Nearest Neighbour (IBK), and Random Forest (RF) for performing the activity detection from the mobile signals. Wei et al. [41] have proposed a new stacked denoising autoencoder (SDAE) based recommender system for cold items. The algorithm employed deep learning and collaborative filtering method to predict the unknown ratings.

In 2018, Li et al. [42] have developed a recommendation algorithm using Weighted Linear Regression Models (WLRRS). The proposed system was put to experiment using the MovieLens dataset and it presented better classification and predictive accuracy. Mezei and Nikou [43] presented a mobile health and wellness recommender system based on fuzzy optimization. It could recommend a collection of actions to be taken by the user to improve the user's health condition. Recommendations were made considering the user's physical activities and preferences. Ayata et al. [44] proposed a music recommendation model based on the user emotions captured through wearable physiological sensors. The emotion detection algorithm employed different machine learning algorithms like SVM, RF, KNN and decision tree (DT) algorithms to predict the emotions from the changing electrical signals gathered from the wearable sensors. Zhao et al. [45] developed a multimodal learning-based, social-aware movie recommender system. The model was able to successfully resolve the sparsity problem of recommender systems. The algorithm developed a heterogeneous network by exploiting the movie-poster image and textual description of each movie based on the social relationships and user ratings.

In 2019, Hammou et al. [46] proposed a Big Data recommendation algorithm capable of handling large scale data. The system employed random forest and matrix factorization through a data partitioning scheme. It was then used for generating recommendations based on user rating and preference for each item. The proposed system outperformed existing systems in terms of accuracy and speed. Zhao et al. [47] have put forward a hybrid initialization method for social network recommender systems. The algorithm employed denoising autoencoder (DAE) neural network-based initialization method (ANNInit) and attribute mapping. Bhaskaran and Santhi [48] have developed a hybrid, trust-based e-learning recommender system using cloud computing. The proposed algorithm was capable of learning online user activities by using the Firefly Algorithm (FA) and K-means clustering. Afolabi and Toivanen [59] have suggested an integrated recommender model based on collaborative filtering. The proposed model "Connected Health for Effective Management of Chronic Diseases", aimed for integrating recommender systems for better decision-making in the process of disease management. He et al. [60] proposed a movie recommender system called "HI2Rec" which explored the usage of collaborative filtering and heterogeneous information for making movie recommendations. The model used the knowledge representation learning approach to embed movie-related information gathered from different sources.

In 2020, Han et al. [49] have proposed one Internet of Things (IoT)-based cancer rehabilitation recommendation system using the Beetle Antennae Search (BAS) algorithm. It presented the patients with a solution for the problem of optimal nutrition program by considering the objective function as the recurrence time. Kang et al. [50] have presented a recommender system for personalized advertisements in Online Broadcasting based on a tree model. Recommendations were generated in real-time by considering the user preferences to minimize the overhead of preference prediction and using a HashMap along with the tree characteristics. Ullah et al. [51] have implemented an image-based service recommendation model for online shopping based random forest and Convolutional Neural Networks (CNN). The model used JPEG coefficients to achieve an accurate prediction rate. Cai et al. [52] proposed a new hybrid recommender model using a many-objective evolutionary algorithm (MaOEA). The proposed algorithm was successful in optimizing the novelty, diversity, and accuracy of recommendations. Esteban et al. [53] have implemented a hybrid multi-criteria recommendation system concerned with students' academic performance, personal interests, and course selection. The system was developed using a Genetic Algorithm (GA) and aimed at helping university students. It combined both course information and student information for increasing system performance and the reliability of the recommendations. Mondal et al. [54] have built a multilayer, graph data model-based doctor recommendation system by exploiting the trust concept between a patient-doctor relationship. The proposed system showed good results in practical applications.

In 2021, Dhelim et al. [55] have developed a personality-based product recommending model using the techniques of meta path discovery and user interest mining. This model showed better results when compared to session-based and deep learning models. Bhalse et al. [56] proposed a web-based movie recommendation system based on collaborative filtering using Singular Value Decomposition (SVD), collaborative filtering and cosine similarity (CS) for addressing the sparsity problem of recommender systems. It suggested a recommendation list by considering the content information of movies. Similarly, to solve both sparsity and cold-start problems Ke et al. [57] proposed a dynamic goods recommendation system based on reinforcement learning. The proposed system was capable of learning from the reduced entropy loss error on real-time applications. Chen et al. [58] have presented a movie recommender model combining various techniques like user interest with category-level representation, neighbour-assisted representation, user interest with latent representation and item-level representation using Feed-forward Neural Network (FNN).

Comparative chronological review

A comparative chronological review to compare the total contributions on various recommender systems in the past 10 years is given in Fig. 7.

This review puts forward a comparison of the number of research works proposed in the domain of recommender systems from the year 2011 to 2021 using various deep learning and machine learning-based approaches. Research articles are categorized based on the recommender system classification framework as shown in Table 5. The articles are ordered according to their year of publication. There are two key concepts: Application fields and techniques used. The application fields of recommender systems are divided into six different fields, viz. entertainment, health, tourism, web/e-commerce, education and social media/others.

Algorithmic categorization, simulation platforms and applications considered for various recommender systems

This section analyses different methods like deep learning, machine learning, clustering and meta-heuristic-based-approaches used in the development of recommender systems. The algorithmic categorization of different recommender systems is given in Fig. 8.

Categorization is done based on content-based, collaborative filtering-based, and optimization-based approaches. In [8], a content-based filtering technique was employed for increasing the ability to trust other agents and for improving the exchange of information by trust degree. In [16], it was applied to enhance the quality of recommendations using the account attributes of the material. It achieved better performance concerning

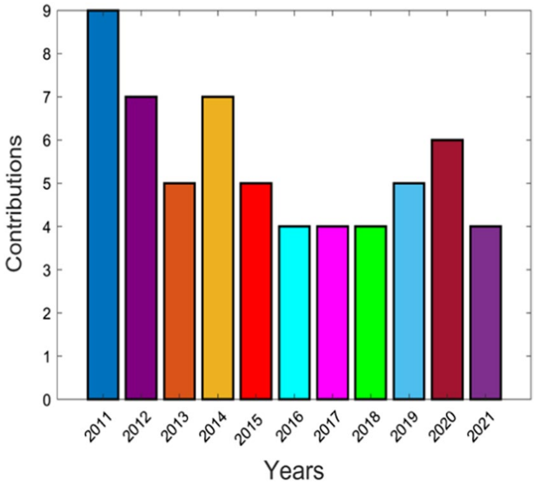

Fig. 7 Comparative chronological review of recommender systems under diverse applications

Table 5 Classification of most relevant research articles

References	Sector/application fields					Techniques used			
	Entertainment	Health	Tourism	Web/e-commerce	Education	Social media/others	Collaborative	Content-based	Hybrid
Castellano et al. [1]				*			*		
Crespo et al. [2]					*		*		*
Lin et al. [3]						*	*		
Wang and Wu [4]					*		*		
García-Crespo et al. [5]			*				*		
Dong et al. [6]						*	*		
Li et al. [7]						*			*
Lorenzi et al. [8]			*					*	
Huang et al. [9]						*	*		
Chen et al. [10]		*					*		
Mohanraj et al. [11]				*					*
Hsu et al. [12]						*			*
Gemmell et al. [13]	*			*		*			*
Choi et al. [14]				*			*		
Garibaldi et al. [15]		*					*		
Salehi and Kmalabadi [16]					*			*	
Aher and Lobo [17]					*		*		
Kardan and Ebrahimi [18]	*			*	*	*		*	
Chang et al. [19]	*						*		
Lucas et al. [20]			*				*		
Niu et al. [21]	*			*			*		
Liu et al. [22]			*						*
Bakshi et al. [23]	*								*
Kim and Shim [24]				*			*		

Table 5 (continued)

References	Sector/application fields					Techniques used			
	Entertainment	Health	Tourism	Web/e-commerce	Education	Social media/ others	Collaborative	Content-based	Hybrid
Wang et al. [25]	*								*
Kolomvatsos et al. [26]	*						*		
Gottschlich et al. [27]						*			*
Torshizi et al. [28]		*							*
Zahálka et al. [29]			*				*		
Sankar et al. [30]						*	*		
Chen et al. [31]	*								*
Wu et al. [32]						*			*
Yeh and Cheng [33]			*						*
Liao et al. [34]				*			*		
Li et al. [35]	*						*		
Wu et al. [36]						*			*
Adeniyi et al. [37]				*			*		
Rawat and Kankanhalli [38]			*			*			*
Yang et al. [39]				*		*	*		
Lee et al. [40]	*						*		
Wei et al. [41]	*						*		
Li et al. [42]	*						*		
Mezei and Nikou [43]		*							*
Ayata et al. [44]		*					*		
Zhao et al. [45]	*						*		
Hammou et al. [46]	*						*		
Zhao et al. [47]	*						*		
Bhaskaran and Santhi [48]					*				*
Afolabi and Toivanen [59]		*					*		

Table 5 (continued)

References	Sector/application fields					Techniques used			
	Entertainment	Health	Tourism	Web/e-commerce	Education	Social media/others	Collaborative	Content-based	Hybrid
He et al. [60]	*						*		
Han et al. [49]		*							*
Kang et al. [50]				*			*		
Ullah et al. [51]				*			*		*
Cai et al. [52]	*								*
Esteban et al. [53]					*				*
Mondal et al. [54]		*					*		
Dhelim et al. [55]						*	*		
Bhalse et al. [56]	*						*		
Ke et al. [57]						*	*		
Chen et al. [58]	*						*		

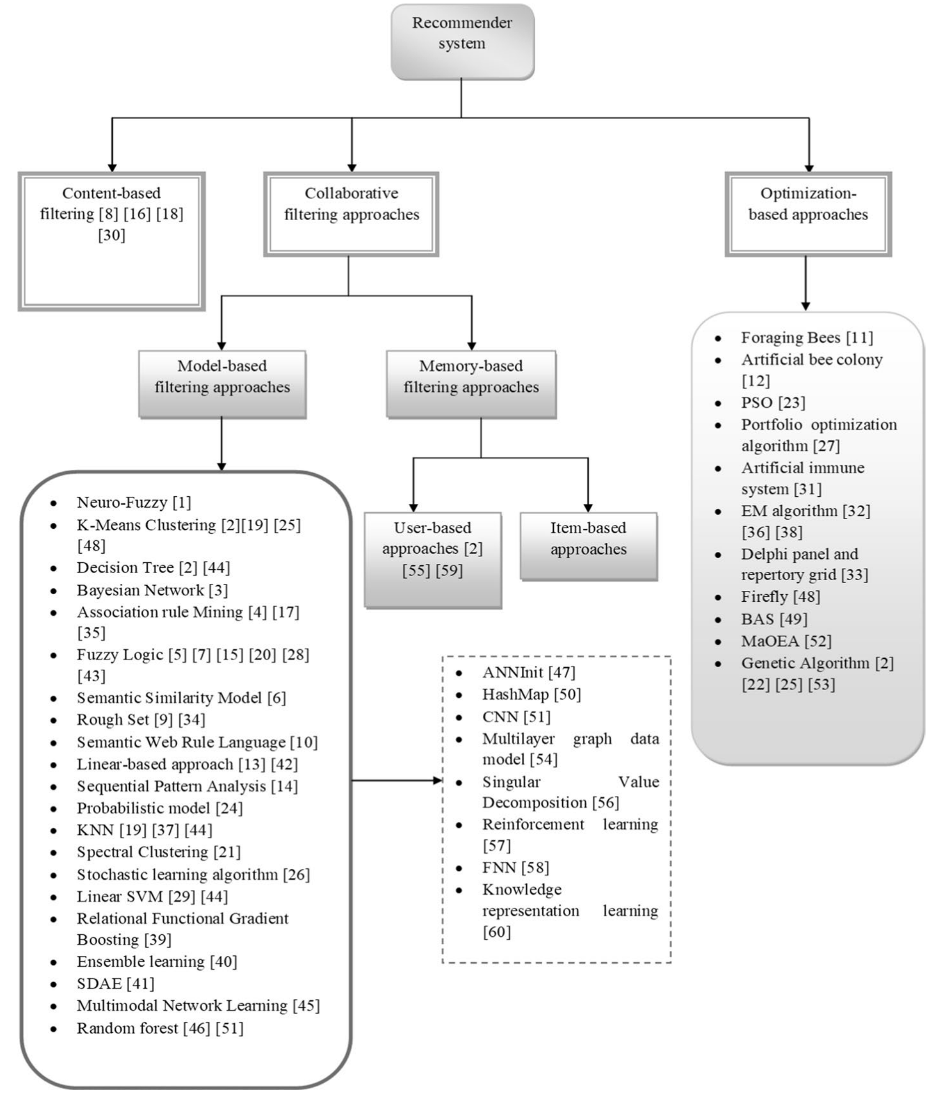
Recommender system
Content-based filtering [8] [16] [18] [30]
Collaborative filtering approaches
Optimization-based approaches
Model-based filtering approaches
Memory-based filtering approaches
User-based approaches [2] [55] [59]
Item-based approaches
Neuro-Fuzzy [1]
K-Means Clustering [2][19] [25] [48]
Decision Tree [2] [44]
Bayesian Network [3]
Association rule Mining [4] [17] [35]
Fuzzy Logic [5] [7] [15] [20] [28] [43]
Semantic Similarity Model [6]
Rough Set [9] [34]
Semantic Web Rule Language [10]
Linear-based approach [13] [42]
Sequential Pattern Analysis [14]
Probabilistic model [24]
KNN [19] [37] [44]
Spectral Clustering [21]
Stochastic learning algorithm [26]
Linear SVM [29] [44]
Relational Functional Gradient Boosting [39]
Ensemble learning [40]
SDAE [41]
Multimodal Network Learning [45]
Random forest [46] [51]
ANNInit [47]
HashMap [50]
CNN [51]
Multilayer graph data model [54]
Singular Value Decomposition [56]
Reinforcement learning [57]
FNN [58]
Knowledge representation learning [60]
Foraging Bees [11]
Artificial bee colony [12]
PSO [23]
Portfolio optimization algorithm [27]
Artificial immune system [31]
EM algorithm [32] [36] [38]
Delphi panel and repertory grid [33]
Firefly [48]
BAS [49]
MaOEA [52]
Genetic Algorithm [2] [22] [25] [53]

Fig. 8 Algorithmic categorization of different recommender systems

with F1-score, recall and precision. In [18], this technique was able to capture the implicit user feedback, increasing the overall accuracy of the proposed model. The content-based filtering in [30] was able to increase the accuracy and performance of a stock recommender system by using the "trust factor" for making decisions.

Different collaborative filtering approaches are utilized in recent studies, which are categorized as follows:

Model-based techniques

Neuro-Fuzzy [1] based technique helps in discovering the association between user categories and item relevance. It is also simple to understand. K-Means Clustering [2, 19, 25, 48] is efficient for large scale datasets. It is simple to implement and gives a fast convergence rate. It also offers automatic recovery from failures. The decision tree [2, 44] technique is easy to interpret. It can be used for solving the classic regression and classification problems in recommender systems. Bayesian Network [3] is a probabilistic technique used to solve classification challenges. It is based on the theory of Bayes theorem and conditional probability. Association Rule Mining (ARM) techniques [4, 17, 35] extract rules for projecting the occurrence of an item by considering the existence of other items in a transaction. This method uses the association rules to create a more suitable representation of data and helps in increasing the model performance and storage efficiency. Fuzzy Logic [5, 7, 15, 20, 28, 43] techniques use a set of flexible rules. It focuses on solving complex real-time problems having an inaccurate spectrum of data. This technique provides scalability and helps in increasing the overall model performance for recommender systems. The semantic similarity [6] technique is used for describing a topological similarity to define the distance among the concepts and terms through ontologies. It measures the similarity information for increasing the efficiency of recommender systems. Rough set [9, 34] techniques use probability distributions for solving the challenges of existing recommender models. Semantic web rule language [10] can efficiently extract the dataset features and increase the model efficiency. Linear programming-based approaches [13, 42] are employed for achieving quality decision making in recommender models. Sequential pattern analysis [14] is applied to find suitable patterns among data items. This helps in increasing model efficiency. The probabilistic model [24] is a famous tool to handle uncertainty in risk computations and performance assessment. It offers better decision-making capabilities. K-nearest neighbours (KNN) [19, 37, 44] technique provides faster computation time, simplicity and ease of interpretation. They are good for classification and regression-based problems and offers more accuracy. Spectral clustering [21] is also called graph clustering or similarity-based clustering, which mainly focuses on reducing the space dimensionality in identifying the dataset items. Stochastic learning algorithm [26] solves the real-time challenges of recommender systems. Linear SVM [29, 44] efficiently solves the high dimensional problems related to recommender systems. It is a memory-efficient method and works well with a large number of samples having relative separation among the classes. This method has been shown to perform well even when new or unfamiliar data is added. Relational Functional Gradient Boosting [39] technique efficiently works on the relational dependency of data, which is useful for statical relational learning for collaborative-based recommender systems. Ensemble learning [40] combines the forecast of two or more models and aims to achieve better performance than any of the single contributing models. It also helps in reducing overfitting problems, which are common in recommender systems.

SDAE [41] is used for learning the non-linear transformations with different filters for finding suitable data. This aids in increasing the performance of recommender models. Multimodal network learning [45] is efficient for multi-modal data, representing a combined representation of diverse modalities. Random forest [46, 51] is a commonly used approach in comparison with other classifiers. It has been shown to increase accuracy when handling big data. This technique is a collection of decision trees to minimize variance through training on diverse data samples. ANNInit [47] is a type of artificial neural network-based technique that has the capability of self-learning and generating efficient results. It is independent of the data type and can learn data patterns automatically. HashMap [50] gives faster access to elements owing to the hashing methodology, which decreases the data processing time and increases the performance of the system. CNN [51] technique can automatically fetch the significant features of a dataset without any supervision. It is a computationally efficient method and provides accurate recommendations. This technique is also simple and fast for implementation. Multilayer graph data model [54] is efficient for real-time applications and minimizes the access time through mapping the correlation as edges among nodes and provides superior performance. Singular Value Decomposition [56] can simplify the input data and increase the efficiency of recommendations by eliminating the noise present in data. Reinforcement learning [57] is efficient for practical scenarios of recommender systems having large data sizes. It is capable of boosting the model performance by increasing the model accuracy even for large scale datasets. FNN [58] is one of the artificial neural network techniques which can learn non-linear and complex relationships between items. It has demonstrated a good performance increase when employed in different recommender systems. Knowledge representation learning [60] systems aim to simplify the model development process by increasing the acquisition efficiency, inferential efficiency, inferential adequacy and representation adequacy. User-based approaches [2, 55, 59] specialize in detecting user-related meta-data which is employed to increase the overall model performance. This technique is more suitable for real-time applications where it can capture user feedback and use it to increase the user experience.

Optimization-based techniques

The Foraging Bees [11] technique enables both functional and combinational optimization for random searching in recommender models. Artificial bee colony [12] is a swarm-based meta-heuristic technique that provides features like faster convergence rate, the ability to handle the objective with stochastic nature, ease for incorporating with other algorithms, usage of fewer control parameters, strong robustness, high flexibility and simplicity. Particle Swarm Optimization [23] is a computation optimization technique that offers better computational efficiency, robustness in control parameters, and is easy and simple to implement in recommender systems. Portfolio optimization algorithm [27] is a subclass of optimization algorithms that find its application in stock investment recommender systems. It works well in real-time and helps in the diversification of the portfolio for maximum profit. The artificial immune system [31]a is computationally intelligent machine learning technique. This technique can learn new patterns in the data and optimize the overall system parameters. Expectation maximization (EM) [32, 36, 38] is an iterative algorithm that guarantees the likelihood of finding the maximum parameters when the input variables are unknown. Delphi panel and repertory grid [33] offers efficient decision making by solving the dimensionality problem and data sparsity issues of recommender systems. The Firefly algorithm (FA) [48] provides fast results and increases recommendation efficiency. It is capable of reducing the number of iterations required to solve specific recommender problems. It also provides both local and global sets of solutions. Beetle Antennae Search (BAS) [49] offers superior search accuracy and maintains less time complexity that promotes the performance of recommendations. Many-objective evolutionary algorithm (MaOEA) [52] is applicable for real-time, multi-objective, search-related recommender systems. The introduction of a local search operator increases the convergence rate and gets suitable results. Genetic

Table 6 Features and challenges of different recommender systems

References	Methodology	Features	Challenges
Castellano et al. [1]	Neuro-fuzzy	High accuracy	Model is not efficient for real-time applications
Crespo et al. [2]	K-means clustering	Solves the problems related to electronic commerce and provides a good user experience	Model is not applicable on huge datasets
Lin et al. [3]	GA, k-means, DT and BN	Increases profit in real-time applications along with product sale volumes	Lacks in operation efficiency
Wang and Wu [4]	Association rule mining	Useful for real-time ubiquitous-learning applications	The model does not consider the learner's behaviour
García-Crespo et al. [5]	Fuzzy logic	High recommendation accuracy and scalability	Not applicable on large datasets
Dong et al. [6]	Semantic similarity model	Solves the issue of service supply ecosystems in a software environment	Low overall performance
Li et al. [7]	Fuzzy linguistic modelling	Good expert recommendations for knowledge management systems	Does not consider the user's past preferences
Lorenzi et al. [8]	Content-based filtering	This model can alleviate the sparsity issue of recommender systems	Low accuracy
Huang et al. [9]	Rough set model	It enhances the accuracy of recommender systems	The model is complex for real-time applications
Chen et al. [10]	Semantic web rule language	The model can efficiently extract the features of the dataset	The model is dependent on domain ontology data
Mohanraj et al. [11]	Foraging Bees algorithm	A self-adaptive model with low computational time	Does not consider any change in user interests
Hsu et al. [12]	Artificial bee colony algorithm	It has attained better accuracy and improved the execution time	The system is limited to the specific application only
Gemmell et al. [13]	Linear-weighted hybrid approach	The model is highly effective and flexible for capitalizing on a robust relationship in different data dimensions	The model is complex and requires high memory
Choi et al. [14]	Sequential pattern analysis	High accuracy and performance	This model cannot be used for large datasets
Garibaldi et al. [15]	Fuzzy rule system	The model integrates variability with a conventional fuzzy inference system	Not suitable for sparse datasets
Salehi [16]	Content-based filtering	Good F1 score, recall and precision values	Precision decreases with the increase of the size of the recommendation sets
Aher and Lobo [17]	K-means and Association rule mining (ARM)	The model is capable of efficiently recommending online courses to new students	Does not consider specific choices of individual users
Kardan and Ebrahimi [18]	ARM	This hybrid model delivers better efficiency when compared with individual models. It also considers implicit user information	The algorithm does not rank the outcome of the mining algorithm
Chang et al. [19]	HFC, K-means clustering and KNN	Improved resource utilization in a cloud environment	This model suffers from hardware resource complications

Table 6 (continued)

References	Methodology	Features	Challenges
Lucas et al. [20]	Associative classification and Fuzzy logic	The model successfully alleviates problems such as gray sheep, cold-start, sparsity, and scalability issues	The model is not suitable for web mining applications
Niu et al. [21]	Spectral clustering	The system attains better performance in terms of recall, precision and F1-score, and delivers a good user experience	Not applicable for low-level features
Liu et al. [22]	Genetic Algorithm	A unique model was developed to save the travelling time of self-driven tourists	The model is platform-dependent, and the recommendations are not user-customizable
Bakshi et al. [23]	Unsupervised learning, PSO	The model addresses the issues of scalability and sparse datasets	The selection of clusters is done arbitrarily
Kim and Shim [24]	Probabilistic model	Good precision, recall and average hit rank	The model is complex
Wang et al. [25]	GA and K-means	High prediction accuracy	The model is not suitable for real-time applications and sparse data
Kolomvatsos et al. [26]	Stochastic decision making	It can efficiently handle multiple requests and save resources and time	The model is not suitable for real-time applications
Gottschlich et al. [27]	Portfolio optimization algorithm	A unique model which considers the crowd's wisdom to maximize a stock investor's portfolio performance	It is a general model which does not consider specific user needs
Torshizi et al. [28]	Fuzzy rule-based systems	A novel model useful for prostate treatment recommendations	The model strongly relies on medical data which lacks certainty
Zahálka et al. [29]	Linear SVM model	An information-rich and interactive venue recommender model capable of connecting tourists	The model has low precision values for some datasets
Sankar et al. [30]	Content-based filtering approach	A simple and accurate model for investment guidance	Hidden relationships among features may result in low accuracy
Chen et al. [31]	Artificial immune system	It employs a new prediction formula to improve the recall, precision, and F1-scores	Suffers from data scalability, cold-start and memory-based issues
Wu et al. [32]	Expectation maximization (EM) -based algorithm	Highly robust in fusing item recommendations for collaborative tagging systems	Low recommendation quality and not suitable for large datasets
Yeh and Cheng [33]	Delphi panel and repertory grid techniques	The model is easy to use and solves the dimensionality problem and data sparsity issue	Not applicable for real-time applications
Liao et al. [34]	Rough set association rule	The model considers dynamic user behaviour	High computation complexity and not fit for other applications
Li et al. [35]	Association rule mining	The model can be applied to many different applications	Performance varies for different applications
Wu et al. [36]	CTR and SMF	This model is simple and robust having high accuracy	Does not consider the social trust among users

Table 6 (continued)

References	Methodology	Features	Challenges
Adeniyi et al. [37]	KNN algorithm	A scalable model with a good precision rate and low computing load	Not applicable for real-time applications
Rawat and Kankanhalli [38]	Image processing	A scalable system capable of learning offline and working in real-time	The system lacks personalization
Yang et al. [39]	Hybrid gradient boosting	Suitable for real-time and large-scale recommender systems. It efficiently reduces false positive predictions	The model suffers from the cold-start problem and requires pre-clustering and feature engineering
Lee et al. [40]	Collaborative ensemble learning	The model can handle dynamic user behaviour captured from mobile signals	Suffers from data sparsity and cold start issues
Wei et al. [41]	SDAE	The model efficiently solves the cold start problem	Requires high computational resources and storage
Li et al. [42]	Weighted linear regression	Low computational complexity and low prediction and modelling time for real-time applications	The model is complex to implement in real large-scale applications
Mezei and Nikou [43]	Fuzzy optimization	Better accuracy and F1-score	Not scalable and cannot adopt other variants of fuzzy sets
Ayata et al. [44]	SVM, RF, DT and KNN	The model can capture real-time user emotion in a dynamic environment	The model is susceptible to the failure of sensors
Zhao et al. [45]	Multimodal network learning	This high-performance model is capable of handling real-world, large-scale datasets	It suffers from high computational costs and premature convergence
Hammou et al. [46]	Matrix factorization and RF	The model reduced the cost functions by efficient learning of parameters to overcome the data sparsity problem	High computational cost and does not consider social contexts of users
Zhao et al. [47]	ANNInit	A scalable model with high prediction accuracy and convergence efficiency	Not suitable for multi-dimension applications
Bhaskaran and Santhi [48]	FA and K-mean clustering	High performance and accuracy with low error and computation time	Cannot be applied to other optimization methods
Afolabi and Toivanen [59]	User-based approaches	A unique system where patients can share recommendations in real-time resulting in better self-care and decision making	Requires a smooth connection between different healthcare services
He et al. [60]	Knowledge representation learning	A high-performance model which alleviates the data sparsity issue and cold start problem	The model is complex
Han et al. [49]	BAS	The model is simple, portable and has high prediction accuracy with a fast convergence rate	BAS algorithm suffers from low search accuracy and limited exploration scope
Kang et al. [50]	Hashmap	Solves the problem of data sparsity and overspecialization	High time complexity

Table 6 (continued)

References	Methodology	Features	Challenges
Ullah et al. [51]	RF and CNN	High efficacy and suitable for real-time product recommendations	It suffers from high time complexity
Cai et al. [52]	MaoEA	High accuracy and applicability in diverse applications	The model lacks stability
Esteban et al. [53]	GA	High performance and reliability for real-time course recommendations	The model lacks user personalization
Mondal et al. [54]	Multilayer graph data model	It provides trust-based results with no requirement of external metadata for calculating the trust factor	The model is only applicable to limited scenarios
Dhelim et al. [55]	User-based approaches	The system can efficiently detect the meta-interest of the user and has a good recall and precision score for cold-start environments	It requires high computational power to decrease response time
Bhalse et al. [56]	SVD, CS and collaborative filtering	It reduces the complexity by reducing the multidimensionality issue	Suffers from sparsity problem and low accuracy
Ke et al. [57]	Reinforcement learning	High prediction accuracy and low entropy loss error	Not suitable for real-world problems
Chen et al. [58]	FNN	The system delivers personalized suggestions by aggregating multiple user interest representations	Suffers from cold-start problem

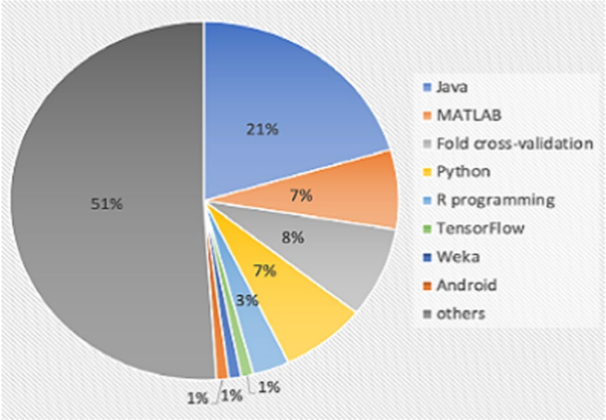

Fig. 9 Simulation platforms used for developing different recommender systems

Algorithm (GA) [2, 22, 25, 53] based techniques are used to solve the multi-objective optimization problems of recommender systems. They employ probabilistic transition rules and have a simpler operation that provides better recommender performance.

Features and challenges

The features and challenges of the existing recommender models are given in Table 6.

Simulation platforms

The various simulation platforms used for developing different recommender systems with different applications are given in Fig. 9.

Here, the Java platform is used in 20% of the contributions, MATLAB is implemented in 7% of the contributions, different fold cross-validation are used in 8% of the contributions, 7% of the contributions are utilized by the python platform, 3% of the contributions employ R-programming and 1% of the contributions are developed by Tensorflow, Weka and Android environments respectively. Other simulation platforms like Facebook, web UI (User Interface), real-time environments, etc. are used in 50% of the contributions. Table 7 describes some simulation platforms commonly used for developing recommender systems.

Application focused and dataset description

This section provides an analysis of the different applications focused on a set of recent recommender systems and their dataset details.

Recent recommender systems were analysed and found that 11% of the contributions are focused on the domain of healthcare, 10% of the contributions are on movie recommender systems, 5% of the contributions come from music recommender systems, 6% of the contributions are focused on e-learning recommender systems, 8% of the contributions are used for online product recommender systems, 3% of the contributions are focused on book recommendations and 1% of the contributions are focused on Job and knowledge management recommender systems.

Table 7 Common simulation platforms and their related information

Simulation platforms/ languages	Platform portability	Web API/GUI integration	Common application domains	Features
Java	***	***	e-learning, health, music, social media, web	Good for developing real-time recommender systems
MATLAB	***	**	Movies, health, music, web	It contains a specific set of classes for recommender systems research
Python	***	***	Health, movies, music, web, social media	It is good for implementing sophisticated machine learning recommender algorithms with high computation requirements
R programming	**	**	Education, news	It is slower compared to Python and MATLAB. It has a high memory requirement
TensorFlow	***	***	Movies, e-learning	Good for implementing unsupervised algorithms
Weka	***	**	e-learning, movies	Excellent for recommender system datasets which require high data pre-processing
Android	*	***	Tourism, web, music	It is excellent for real-time recommender systems but has limited processing capacity

Poor: *; Moderate: **; Good: ***

Table 8 Application focused and dataset description in different recommender systems

References	Application	Dataset description
Castellano et al. [1]	Usage-based Web recommender system	Datasets consist of real-world data gathered from the log-files of a commonly visited website of the successful Japanese film: Dragon Ball Dataset size: 434 input–output samples
Crespo et al. [2]	Intelligent electronic books	–
Lin et al. [3]	Automatic vending machines	Data from nine vending machines were collected from two locations
Wang and Wu [4]	Ubiquitous-learning system	Botanical database Dataset size: 80 knowledge units
García-Crespo et al. [5]	Hotels	Data collected from 10 hotels in Mallorca (Spain)
Dong et al. [6]	Business and software environment	Service metadata Dataset size: 1000
Li et al. [7]	Knowledge management systems for the aircraft industry	–
Lorenzi et al. [8]	Tourism domain	A private dataset was collected from a travel agency Dataset size: 35 test cases
Huang et al. [9]	Context-aware recommender system for movie recommendation	Set of real-world movie rating data Dataset size:1048 ratings from 116 users
Chen et al. [10]	Anti-diabetic drugs selection	Database collected from a professional community of America based physicians Dataset size: 20 patients
Mohanraj et al. [11]	Online recommender system	Data sourced from an intranet website
Hsu et al. [12]	Recommendations for auxiliary material	Facebook data Dataset size: 10 datasets each having 200 to 2000 Facebook posts
Gemmell et al. [13]	Resource recommender for social annotation systems	Data sourced from: Bibsonomy (size: 13,909 annotations), Citeulike (size: 105,873 annotations), movielens (size: 35,366 annotations), Delicious (size: 720,788 annotations), Amazon (size: 498,217 annotations) and lastfm (size: 172,177 annotations)
Choi et al. [14]	Online-product recommender system	Data was collected from a major online shopping mall in Korea Dataset size: 16,486 transactions
Garibaldi et al. [15]	Breast cancer treatment recommendation	Data was collected from a hospital in the United Kingdom Dataset size: 1310 patients
Salehi and Kmalabadi [16]	E-learning material recommender system	Open-source data gathered from a course management system Dataset size: 16,345 records
Aher and Lobo [17]	E-learning recommender system	Open-source data gathered from a course management system
Kardan and Ebrahimi [18]	A content recommender system for asynchronous discussion groups	MetaFilter dataset (size: 229,401 threads) and Yahoo Answers dataset (size: 7,941,404 threads)
Chang et al. [19]	A cloud-based intelligent TV program	User profile database Dataset size: 300,000 user profiles
Lucas et al. [20]	Tourism recommender system	Data was collected from a personalized tourist-aiding system Dataset size: 241
Niu et al. [21]	Video recommender system	Metadata of videos Dataset size: 362 videos
Liu et al. [22]	Recommender system for self-driven tourists	Data collected from Google Maps

Table 8 (continued)

References	Application	Dataset description
Bakshi et al. [23]	E-commerce recommender system	MovieLens dataset of 100,000 ratings provided by 943 users on 1682 movies
Kim and Shim [24]	Twitter	Data fetched from Twitter Dataset size: 12,098,339 tweet messages
Wang et al. [25]	Movie recommender system	MovieLens dataset of 100,000 ratings by 943 users on 1682 movies
Kolomvatsos et al. [26]	Music and book recommender system	An open-source song dataset Dataset size: 1167 songs
Gottschlich et al. [27]	Stock investment recommender system	Data were collected from the 30 largest public companies in Germany
Torshizi et al. [28]	Benign prostatic hyperplasia	44 patients' data were collected from a Medical University
Zahálka et al. [29]	Venue recommender	New York dataset (size: 7,246 venues and 1,072,181 items), and Amsterdam dataset (size: 693 venues and 55,990 items)
Sankar et al. [30]	Stock recommender system	Data were collected from 17 mutual funds
Chen et al. [31]	Movie recommender system	Data was collected from the MovieLens dataset of 100,000 ratings collected from 943 users on 1682 movies and eachmovie dataset with 2,811,983 ratings from 61,265 users on 1623 movies
Wu et al. [32]	Item recommender system	Delicious (Dataset size: 69,223 items), Lastfm (Dataset size: 12,523 items) and CiteULike (Dataset size: 7,188 items)
Yeh and Cheng [33]	Tourist management	Tourist related data was collected from 457 users
Liao et al. [34]	e-commerce recommender system	Data was collected from many online shopping platforms
Li et al. [35]	Movie recommender system	Netflix Prize dataset (size: 100,480,507 ratings from 480,189 customers on 17,770 movies)
Wu et al. [36]	Social media recommender system	Delicious (Dataset size: 45,419 items) and Lastfm (Dataset size: 12,523 items)
Adeniyi et al. [37]	Web usage-based recommender system	Web server data was collected from 5285 web pages
Rawat and Kankanhalli [38]	Mobile photography	Flickr API Dataset size: 67,000 images
Yang et al. [39]	Job recommender system	4 months' data were extracted from the job posting and job application records
Lee et al. [40]	Music streaming recommender system	An open-source song dataset
Wei et al. [41]	Cold start items recommender system	Large Netflix dataset with 100 million ratings for 17,770 movies given by 480,189 users
Li et al. [42]	Diverse applications	MovieLens dataset of 1 million ratings on 3952 movies from 6040 users
Mezei and Nikou [43]	Health and wellness recommender systems	Publicly available medical dataset Dataset size: 20,560 data points
Ayata et al. [44]	Music recommender system	Multimodal DEAP emotion database Dataset size: 32 items
Zhao et al. [45]	Social-aware movie recommender system	A Chinese movie recommending website has 99,641 ratings on 59,695 movies collected from 4,242 users
Hammou et al. [46]	Big Data recommender system	Two MovieLens datasets of size 10 million and 20 million ratings respectively, and Yelp dataset with 6,685,900 ratings
Zhao et al. [47]	Social network recommender system	Two MovieLens datasets of size 100,000 and 1 million ratings respectively

Table 8 (continued)

References	Application	Dataset description
Bhaskaran and Santhi [48]	E-learning and cloud computing recommender system	A public Book-Crossing dataset with 2,78,858 users providing 1,149,780 ratings on 271,379 books
Afolabi and Toivanen [59]	Chronic disease management recommendation	Health-related data collected from devices and sensors
He et al. [60]	Movie recommender system	MovieLens dataset Dataset size: 1,000,000 movie ratings
Han et al. [49]	Cancer rehabilitation recommender system	Cancer rehabilitation medical data
Kang et al. [50]	Advertisement recommender system for online broadcasting	Amazon product data has data from 1557 users, 79,344 reviews, and 55,101 items
Ullah et al. [51]	Image-based service recommendation	Amazon product image data Dataset size: 3.5 million items
Cai et al. [52]	General-purpose recommender system	MovieLens dataset Dataset size: 1,000,000 movie ratings
Esteban et al. [53]	University course selection recommendation	Student data was collected from a university Dataset size: 2500 items
Mondal et al. [54]	Doctor recommender system	Neo4j NoSQL graph database, and patient-doctor information collected from different hospitals Dataset size: 5300 patient records
Dhelim et al. [55]	Product recommender for online shopping	Integrated dataset of a product recommendation system and a social network platform Dataset size: 6230 items
Bhalse et al. [56]	Movie recommender system	MovieLens dataset containing 100,837 ratings applied over 9743 movies
Ke et al. [57]	Cross-platform dynamic goods recommender system	Data was collected using a cross-platform data collection API Dataset size: 100 users
Chen et al. [58]	Video recommender system	MovieLens dataset of size 10 million items and micro-video dataset of size 1.7 million items

5% of the contributions concentrated on social network recommender systems, 10% of the contributions are focused on tourist and hotels recommender systems, 6% of the contributions are employed for stock recommender systems, and 3% of the contributions contributed for video recommender systems. The remaining 12% of contributions are miscellaneous recommender systems like Twitter, venue-based recommender systems, etc. Similarly, different datasets are gathered for recommender systems based on their application types. A detailed description is provided in Table 8.

Performance analysis of state-of-art recommender systems

The performance evaluation metrics used for the analysis of different recommender systems is depicted in Table 9. From the set of research works, 35% of the works use recall measure, 16% of the works employ Mean Absolute Error (MAE), 11% of the works take Root Mean Square Error (RMSE), 41% of the papers consider precision, 30% of the contributions analyse F1-measure, 31% of the works apply accuracy and

Table 9 Performance analysis of different recommender systems

References	Recall	MAE	RMSE	Precision	F1-measure	Accuracy	Miscellaneous measures
Castellano et al. [1]		–	–			–	–
Crespo et al. [2]	–	–	–	–	–	–	Total interaction time
Lin et al. [3]	–	–	–	–	–		Gain ratio
Wang and Wu [4]	–	–	–	–	–	–	Statistical analysis, T-test
García-Crespo et al. [5]		–	–			–	–
Dong et al. [6]		–	–			–	Harmonic mean, fallout
Li et al. [7]		–	–			–	–
Lorenzi et al. [8]	–	–	–	–	–	–	T-test
Huang et al. [9]			–	–	–	–	–
Chen et al. [10]	–	–	–		–		–
Mohanraj et al. [11]	–	–	–	–	–		–
Hsu et al. [12]	–	–	–	–	–	–	Average execution times
Gemmell et al. [13]		–	–		–	–	α analysis
Choi et al. [14]		–	–				–
Garibaldi et al. [15]	–	–	–	–	–	–	Confusion matrix
Salehi and Kmalabadi [16]		–	–			–	–
Aher and Lobo [17]	–	–	–	–	–	–	Space and time complexity
Kardan and Ebrahimi [18]			–				–
Chang et al. [19]	–	–	–	–	–	–	Processing time
Lucas et al. [20]	–	–	–	–	–	–	Empirical analysis
Niu et al. [21]		–	–			–	Click-through rate, session length and subjective feedbacks
Liu et al. [22]	–	–	–	–	–	–	Total visiting time and the total overtime
Bakshi et al. [23]	–			–	–	–	–
Kim and Shim [24]		–	–		–	–	–
Wang et al. [25]			–	–		–	–
Kolomvatsos et al. [26]	–		–	–	–	–	–
Gottschlich et al. [27]	–	–	–	–	–	–	Standard deviation, Sharpe ratio
Torshizi et al. [28]	–	–	–	–	–	–	Cohen kappa statistic
Zahálka et al. [29]		–	–		–	–	–
Sankar et al. [30]	–	–	–	–	–		Degree centrality, Eigen vector centrality, average normal gain
Chen et al. [31]			–			–	–
Wu et al. [32]	–	–	–	–	–		Inner-diversity, inter-diversity, and novelty
Yeh and Cheng [33]	–	–	–		–	–	–
Liao et al. [34]	–	–	–	–	–	–	Level of confidence
Li et al. [35]	–					–	–
Wu et al. [36]		–	–	–	–		–
Adeniyi et al. [37]	–	–	–		–		Euclidean distance
Rawat and Kankanhalli [38]	–	–	–		–	–	–
Yang et al. [39]		–	–		–	–	–
Po et al. [40]	–	–					–

Table 9 (continued)

References	Recall	MAE	RMSE	Precision	F1-measure	Accuracy	Miscellaneous measures
Wei et al. [41]	–	–		–	–	–	–
Li et al. [42]	–						–
Mezei and Nikou [43]	–	–	–	–	–		Specificity, sensitivity, cohen's κ
Ayata et al. [44]	–	–	–	–	–		–
Zhao et al. [45]	–	–	–		–	–	–
Hammou et al. [46]	–			–	–	–	–
Zhao et al. [47]	–	–		–	–		Convergence speed
Bhaskaran and Santhi [48]		–		–			Execution speed
Afolabi and Toivanen [59]	–	–	–	–	–	–	Timeliness
He et al. [60]		–	–			–	Normalized discounted cumulative gain
Han et al. [49]	–	–	–	–	–		Recovery time, absolute error, recurrence time
Kang et al. [50]		–	–				–
Ullah et al. [51]		–	–			–	–
Cai et al. [52]	–	–	–	–	–		Diversity, and novelty
Esteban et al. [53]	–		–	–	–	–	Time complexity, normalized discounted cumulative gain, and execution time
Mondal et al. [54]	–	–	–	–	–	–	Trust factor
Dhelim et al. [55]		–	–			–	–
Bhalse et al. [56]	–	–	–	–	–	–	–
Ke et al. [57]	–	–	–	–	–		–
Chen et al. [58]	–	–	–	–	–	–	Area under curve (AUC)

6% of the works employ coverage measure to validate the performance of the recommender systems. Moreover, some additional measures are also considered for validating the performance in a few applications.

Research gaps and challenges

In the recent decade, recommender systems have performed well in solving the problem of information overload and has become the more appropriate tool for multiple areas such as psychology, mathematics, computer science, etc. [80]. However, current recommender systems face a variety of challenges which are stated as follows, and discussed below:

- Deployment challenges such as cold start, scalability, sparsity, etc. are already discussed in Sect. 3.
- Challenges faced when employing different recommender algorithms for different applications.
- Challenges in collecting implicit user data
- Challenges in handling real-time user feedback.
- Challenges faced in choosing the correct implementation techniques.

- Challenges faced in measuring system performance.
- Challenges in implementing recommender system for diverse applications.

Numerous recommender algorithms have been proposed on novel emerging dimensions which focus on addressing the existing limitations of recommender systems. A good recommender system must increase the recommendation quality based on user preferences. However, a specific recommender algorithm is not always guaranteed to perform equally for different applications. This encourages the possibility of employing different recommender algorithms for different applications, which brings along a lot of challenges. There is a need for more research to alleviate these challenges. Also, there is a large scope of research in recommender applications that incorporate information from different interactive online sites like Facebook, Twitter, shopping sites, etc. Some other areas for emerging research may be in the fields of knowledge-based recommender systems, methods for seamlessly processing implicit user data and handling real-time user feedback to recommend items in a dynamic environment.

Some of the other research areas like deep learning-based recommender systems, demographic filtering, group recommenders, cross-domain techniques for recommender systems, and dimensionality reduction techniques are also further required to be studied [83]. Deep learning-based recommender systems have recently gained much popularity. Future research areas in this field can integrate the well-performing deep learning models with new variants of hybrid meta-heuristic approaches.

During this review, it was observed that even though recent recommender systems have demonstrated good performance, there is no single standardized criteria or method which could be used to evaluate the performance of all recommender systems. System performance is generally measured by different evaluation matrices which makes it difficult to compare. The application of recommender systems in real-time applications is growing. User satisfaction and personalization play a very important role in the success of such recommender systems. There is a need for some new evaluation criteria which can evaluate the level of user satisfaction in real-time. New research should focus on capturing real-time user feedback and use the information to change the recommendation process accordingly. This will aid in increasing the quality of recommendations.

Conclusion and future scope

Recommender systems have attracted the attention of researchers and academicians. In this paper, we have identified and prudently reviewed research papers on recommender systems focusing on diverse applications, which were published between 2011 and 2021. This review has gathered diverse details like different application fields, techniques used, simulation tools used, diverse applications focused, performance metrics, datasets used, system features, and challenges of different recommender systems. Further, the research gaps and challenges were put forward to explore the future research perspective on recommender systems. Overall, this paper provides a comprehensive understanding of the trend of recommender systems-related research and to provides researchers with insight and future direction on recommender systems. The results of this study have several practical and significant implications:

- Based on the recent-past publication rates, we feel that the research of recommender systems will significantly grow in the future.
- A large number of research papers were identified in movie recommendations, whereas health, tourism and education-related recommender systems were identified in very few numbers. This is due to the availability of movie datasets in the public domain. Therefore, it is necessary to develop datasets in other fields also.
- There is no standard measure to compute the performance of recommender systems. Among 60 papers, 21 used recall, 10 used MAE, 25 used precision, 18 used F1-measure, 19 used accuracy and only 7 used RMSE to calculate system performance. Very few systems were found to excel in two or more matrices.
- Java and Python (with a combined contribution of 27%) are the most common programming languages used to develop recommender systems. This is due to the availability of a large number of standard java and python libraries which aid in the development process.
- Recently a large number of hybrid and optimizations techniques are being proposed for recommender systems. The performance of a recommender system can be greatly improved by applying optimization techniques.
- There is a large scope of research in using neural networks and deep learning-based methods for developing recommender systems. Systems developed using these methods are found to achieve high-performance accuracy.

This research will provide a guideline for future research in the domain of recommender systems. However, this research has some limitations. Firstly, due to the limited amount of manpower and time, we have only reviewed papers published in journals focusing on computer science, management and medicine. Secondly, we have reviewed only English papers. New research may extend this study to cover other journals and non-English papers. Finally, this review was conducted based on a search on only six descriptors: "Recommender systems", "Recommendation systems", "Movie Recommend*", "Music Recommend*", "Personalized Recommend*" and "Hybrid Recommend*". Research papers that did not include these keywords were not considered. Future research can include adding some additional descriptors and keywords for searching. This will allow extending the research to cover more diverse articles on recommender systems.

Acknowledgements

We thank our colleagues from Assam Down Town University who provided insight and expertise that greatly assisted this research, although they may not agree with all the interpretations and conclusions of this paper.

Author contributions

DR carried out the review study and analysis of the existing algorithms in the literature. MD has been involved in drafting the manuscript or revising it critically for important intellectual content. Both authors read and approved the final manuscript.

Funding

No funding was received to assist with the preparation of this manuscript.

Availability of data and materials

Not applicable.

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication
Not applicable.

Competing interests
On behalf of all authors, the corresponding author states that there is no conflict of interest.

Received: 4 October 2021 Accepted: 28 March 2022
Published online: 03 May 2022

References

1. Castellano G, Fanelli AM, Torsello MA. NEWER: A system for neuro-fuzzy web recommendation. Appl Soft Comput. 2011;11:793–806.
2. Crespo RG, Martínez OS, Lovelle JMC, García-Bustelo BCP, Gayo JEL, Pablos PO. Recommendation system based on user interaction data applied to intelligent electronic books. Computers Hum Behavior. 2011;27:1445–9.
3. Lin FC, Yu HW, Hsu CH, Weng TC. Recommendation system for localized products in vending machines. Expert Syst Appl. 2011;38:9129–38.
4. Wang SL, Wu CY. Application of context-aware and personalized recommendation to implement an adaptive ubiquitous learning system. Expert Syst Appl. 2011;38:10831–8.
5. García-Crespo Á, López-Cuadrado JL, Colomo-Palacios R, González-Carrasco I, Ruiz-Mezcua B. Sem-Fit: A semantic based expert system to provide recommendations in the tourism domain. Expert Syst Appl. 2011;38:13310–9.
6. Dong H, Hussain FK, Chang E. A service concept recommendation system for enhancing the dependability of semantic service matchmakers in the service ecosystem environment. J Netw Comput Appl. 2011;34:619–31.
7. Li M, Liu L, Li CB. An approach to expert recommendation based on fuzzy linguistic method and fuzzy text classification in knowledge management systems. Expert Syst Appl. 2011;38:8586–96.
8. Lorenzi F, Bazzan ALC, Abel M, Ricci F. Improving recommendations through an assumption-based multiagent approach: An application in the tourism domain. Expert Syst Appl. 2011;38:14703–14.
9. Huang Z, Lu X, Duan H. Context-aware recommendation using rough set model and collaborative filtering. Artif Intell Rev. 2011;35:85–99.
10. Chen RC, Huang YH, Bau CT, Chen SM. A recommendation system based on domain ontology and SWRL for antidiabetic drugs selection. Expert Syst Appl. 2012;39:3995–4006.
11. Mohanraj V, Chandrasekaran M, Senthilkumar J, Arumugam S, Suresh Y. Ontology driven bee's foraging approach based self-adaptive online recommendation system. J Syst Softw. 2012;85:2439–50.
12. Hsu CC, Chen HC, Huang KK, Huang YM. A personalized auxiliary material recommendation system based on learning style on facebook applying an artificial bee colony algorithm. Comput Math Appl. 2012;64:1506–13.
13. Gemmell J, Schimoler T, Mobasher B, Burke R. Resource recommendation in social annotation systems: A linear-weighted hybrid approach. J Comput Syst Sci. 2012;78:1160–74.
14. Choi K, Yoo D, Kim G, Suh Y. A hybrid online-product recommendation system: Combining implicit rating-based collaborative filtering and sequential pattern analysis. Electron Commer Res Appl. 2012;11:309–17.
15. Garibaldi JM, Zhou SM, Wang XY, John RI, Ellis IO. Incorporation of expert variability into breast cancer treatment recommendation in designing clinical protocol guided fuzzy rule system models. J Biomed Inform. 2012;45:447–59.
16. Salehi M, Kmalabadi IN. A hybrid attribute–based recommender system for e–learning material recommendation. IERI Procedia. 2012;2:565–70.
17. Aher SB, Lobo LMRJ. Combination of machine learning algorithms for recommendation of courses in e-learning System based on historical data. Knowl-Based Syst. 2013;51:1–14.
18. Kardan AA, Ebrahimi M. A novel approach to hybrid recommendation systems based on association rules mining for content recommendation in asynchronous discussion groups. Inf Sci. 2013;219:93–110.
19. Chang JH, Lai CF, Wang MS, Wu TY. A cloud-based intelligent TV program recommendation system. Comput Electr Eng. 2013;39:2379–99.
20. Lucas JP, Luz N, Moreno MN, Anacleto R, Figueiredo AA, Martins C. A hybrid recommendation approach for a tourism system. Expert Syst Appl. 2013;40:3532–50.
21. Niu J, Zhu L, Zhao X, Li H. Affivir: An affect-based Internet video recommendation system. Neurocomputing. 2013;120:422–33.
22. Liu L, Xu J, Liao SS, Chen H. A real-time personalized route recommendation system for self-drive tourists based on vehicle to vehicle communication. Expert Syst Appl. 2014;41:3409–17.
23. Bakshi S, Jagadev AK, Dehuri S, Wang GN. Enhancing scalability and accuracy of recommendation systems using unsupervised learning and particle swarm optimization. Appl Soft Comput. 2014;15:21–9.
24. Kim Y, Shim K. TWILITE: A recommendation system for twitter using a probabilistic model based on latent Dirichlet allocation. Inf Syst. 2014;42:59–77.
25. Wang Z, Yu X, Feng N, Wang Z. An improved collaborative movie recommendation system using computational intelligence. J Vis Lang Comput. 2014;25:667–75.
26. Kolomvatsos K, Anagnostopoulos C, Hadjiefthymiades S. An efficient recommendation system based on the optimal stopping theory. Expert Syst Appl. 2014;41:6796–806.
27. Gottschlich J, Hinz O. A decision support system for stock investment recommendations using collective wisdom. Decis Support Syst. 2014;59:52–62.
28. Torshizi AD, Zarandi MHF, Torshizi GD, Eghbali K. A hybrid fuzzy-ontology based intelligent system to determine level of severity and treatment recommendation for benign prostatic hyperplasia. Comput Methods Programs Biomed. 2014;113:301–13.
29. Zahálka J, Rudinac S, Worring M. Interactive multimodal learning for venue recommendation. IEEE Trans Multimedia. 2015;17:2235–44.

30. Sankar CP, Vidyaraj R, Kumar KS. Trust based stock recommendation system – a social network analysis approach. Procedia Computer Sci. 2015;46:299–305.
31. Chen MH, Teng CH, Chang PC. Applying artificial immune systems to collaborative filtering for movie recommendation. Adv Eng Inform. 2015;29:830–9.
32. Wu H, Pei Y, Li B, Kang Z, Liu X, Li H. Item recommendation in collaborative tagging systems via heuristic data fusion. Knowl-Based Syst. 2015;75:124–40.
33. Yeh DY, Cheng CH. Recommendation system for popular tourist attractions in Taiwan using delphi panel and repertory grid techniques. Tour Manage. 2015;46:164–76.
34. Liao SH, Chang HK. A rough set-based association rule approach for a recommendation system for online consumers. Inf Process Manage. 2016;52:1142–60.
35. Li H, Cui J, Shen B, Ma J. An intelligent movie recommendation system through group-level sentiment analysis in microblogs. Neurocomputing. 2016;210:164–73.
36. Wu H, Yue K, Pei Y, Li B, Zhao Y, Dong F. Collaborative topic regression with social trust ensemble for recommendation in social media systems. Knowl-Based Syst. 2016;97:111–22.
37. Adeniyi DA, Wei Z, Yongquan Y. Automated web usage data mining and recommendation system using K-Nearest Neighbor (KNN) classification method. Appl Computing Inform. 2016;12:90–108.
38. Rawat YS, Kankanhalli MS. ClickSmart: A context-aware viewpoint recommendation system for mobile photography. IEEE Trans Circuits Syst Video Technol. 2017;27:149–58.
39. Yang S, Korayem M, Aljadda K, Grainger T, Natarajan S. Combining content-based and collaborative filtering for job recommendation system: A cost-sensitive Statistical Relational Learning approach. Knowl-Based Syst. 2017;136:37–45.
40. Lee WP, Chen CT, Huang JY, Liang JY. A smartphone-based activity-aware system for music streaming recommendation. Knowl-Based Syst. 2017;131:70–82.
41. Wei J, He J, Chen K, Zhou Y, Tang Z. Collaborative filtering and deep learning based recommendation system for cold start items. Expert Syst Appl. 2017;69:29–39.
42. Li C, Wang Z, Cao S, He L. WLRRS: A new recommendation system based on weighted linear regression models. Comput Electr Eng. 2018;66:40–7.
43. Mezei J, Nikou S. Fuzzy optimization to improve mobile health and wellness recommendation systems. Knowl-Based Syst. 2018;142:108–16.
44. Ayata D, Yaslan Y, Kamasak ME. Emotion based music recommendation system using wearable physiological sensors. IEEE Trans Consum Electron. 2018;64:196–203.
45. Zhao Z, Yang Q, Lu H, Weninger T. Social-aware movie recommendation via multimodal network learning. IEEE Trans Multimedia. 2018;20:430–40.
46. Hammou BA, Lahcen AA, Mouline S. An effective distributed predictive model with matrix factorization and random forest for big data recommendation systems. Expert Syst Appl. 2019;137:253–65.
47. Zhao J, Geng X, Zhou J, Sun Q, Xiao Y, Zhang Z, Fu Z. Attribute mapping and autoencoder neural network based matrix factorization initialization for recommendation systems. Knowl-Based Syst. 2019;166:132–9.
48. Bhaskaran S, Santhi B. An efficient personalized trust based hybrid recommendation (TBHR) strategy for e-learning system in cloud computing. Clust Comput. 2019;22:1137–49.
49. Han Y, Han Z, Wu J, Yu Y, Gao S, Hua D, Yang A. Artificial intelligence recommendation system of cancer rehabilitation scheme based on IoT technology. IEEE Access. 2020;8:44924–35.
50. Kang S, Jeong C, Chung K. Tree-based real-time advertisement recommendation system in online broadcasting. IEEE Access. 2020;8:192693–702.
51. Ullah F, Zhang B, Khan RU. Image-based service recommendation system: A JPEG-coefficient RFs approach. IEEE Access. 2020;8:3308–18.
52. Cai X, Hu Z, Zhao P, Zhang W, Chen J. A hybrid recommendation system with many-objective evolutionary algorithm. Expert Syst Appl. 2020. https://doi.org/10.1016/j.eswa.2020.113648.
53. Esteban A, Zafra A, Romero C. Helping university students to choose elective courses by using a hybrid multi-criteria recommendation system with genetic optimization. Knowledge-Based Syst. 2020;194:105385.
54. Mondal S, Basu A, Mukherjee N. Building a trust-based doctor recommendation system on top of multilayer graph database. J Biomed Inform. 2020;110:103549.
55. Dhelim S, Ning H, Aung N, Huang R, Ma J. Personality-aware product recommendation system based on user interests mining and metapath discovery. IEEE Trans Comput Soc Syst. 2021;8:86–98.
56. Bhalse N, Thakur R. Algorithm for movie recommendation system using collaborative filtering. Materials Today: Proceedings. 2021. https://doi.org/10.1016/j.matpr.2021.01.235.
57. Ke G, Du HL, Chen YC. Cross-platform dynamic goods recommendation system based on reinforcement learning and social networks. Appl Soft Computing. 2021;104:107213.
58. Chen X, Liu D, Xiong Z, Zha ZJ. Learning and fusing multiple user interest representations for micro-video and movie recommendations. IEEE Trans Multimedia. 2021;23:484–96.
59. Afolabi AO, Toivanen P. Integration of recommendation systems into connected health for effective management of chronic diseases. IEEE Access. 2019;7:49201–11.
60. He M, Wang B, Du X. HI2Rec: Exploring knowledge in heterogeneous information for movie recommendation. IEEE Access. 2019;7:30276–84.
61. Bobadilla J, Serradilla F, Hernando A. Collaborative filtering adapted to recommender systems of e-learning. Knowl-Based Syst. 2009;22:261–5.
62. Russell S, Yoon V. Applications of wavelet data reduction in a recommender system. Expert Syst Appl. 2008;34:2316–25.
63. Campos LM, Fernández-Luna JM, Huete JF. A collaborative recommender system based on probabilistic inference from fuzzy observations. Fuzzy Sets Syst. 2008;159:1554–76.
64. Funk M, Rozinat A, Karapanos E, Medeiros AKA, Koca A. In situ evaluation of recommender systems: Framework and instrumentation. Int J Hum Comput Stud. 2010;68:525–47.

65. Porcel C, Moreno JM, Herrera-Viedma E. A multi-disciplinar recommender system to advice research resources in University Digital Libraries. Expert Syst Appl. 2009;36:12520–8.
66. Bobadilla J, Serradilla F, Bernal J. A new collaborative filtering metric that improves the behavior of recommender systems. Knowl-Based Syst. 2010;23:520–8.
67. Ochi P, Rao S, Takayama L, Nass C. Predictors of user perceptions of web recommender systems: How the basis for generating experience and search product recommendations affects user responses. Int J Hum Comput Stud. 2010;68:472–82.
68. Olmo FH, Gaudioso E. Evaluation of recommender systems: A new approach. Expert Syst Appl. 2008;35:790–804.
69. Zhen L, Huang GQ, Jiang Z. An inner-enterprise knowledge recommender system. Expert Syst Appl. 2010;37:1703–12.
70. Göksedef M, Gündüz-Öğüdücü S. Combination of web page recommender systems. Expert Syst Appl. 2010;37(4):2911–22.
71. Shao B, Wang D, Li T, Ogihara M. Music recommendation based on acoustic features and user access patterns. IEEE Trans Audio Speech Lang Process. 2009;17:1602–11.
72. Shin C, Woo W. Socially aware tv program recommender for multiple viewers. IEEE Trans Consum Electron. 2009;55:927–32.
73. Lopez-Carmona MA, Marsa-Maestre I, Perez JRV, Alcazar BA. Anegsys: An automated negotiation based recommender system for local e-marketplaces. IEEE Lat Am Trans. 2007;5:409–16.
74. Yap G, Tan A, Pang H. Discovering and exploiting causal dependencies for robust mobile context-aware recommenders. IEEE Trans Knowl Data Eng. 2007;19:977–92.
75. Meo PD, Quattrone G, Terracina G, Ursino D. An XML-based multiagent system for supporting online recruitment services. IEEE Trans Syst Man Cybern. 2007;37:464–80.
76. Khusro S, Ali Z, Ullah I. Recommender systems: Issues, challenges, and research opportunities. Inform Sci Appl. 2016. https://doi.org/10.1007/978-981-10-0557-2_112.
77. Blanco-Fernandez Y, Pazos-Arias JJ, Gil-Solla A, Ramos-Cabrer M, Lopez-Nores M. Providing entertainment by content-based filtering and semantic reasoning in intelligent recommender systems. IEEE Trans Consum Electron. 2008;54:727–35.
78. Isinkaye FO, Folajimi YO, Ojokoh BA. Recommendation systems: Principles, methods and evaluation. Egyptian Inform J. 2015;16:261–73.
79. Yoshii K, Goto M, Komatani K, Ogata T, Okuno HG. An efficient hybrid music recommender system using an incrementally trainable probabilistic generative model. IEEE Trans Audio Speech Lang Process. 2008;16:435–47.
80. Wei YZ, Moreau L, Jennings NR. Learning users' interests by quality classification in market-based recommender systems. IEEE Trans Knowl Data Eng. 2005;17:1678–88.
81. Bjelica M. Towards TV recommender system: experiments with user modeling. IEEE Trans Consum Electron. 2010;56:1763–9.
82. Setten MV, Veenstra M, Nijholt A, Dijk BV. Goal-based structuring in recommender systems. Interact Comput. 2006;18:432–56.
83. Adomavicius G, Tuzhilin A. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Trans Knowl Data Eng. 2005;17:734–49.
84. Symeonidis P, Nanopoulos A, Manolopoulos Y. Providing justifications in recommender systems. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans. 2009;38:1262–72.
85. Zhan J, Hsieh C, Wang I, Hsu T, Liau C, Wang D. Privacy preserving collaborative recommender systems. IEEE Trans Syst Man Cybernet. 2010;40:472–6.
86. Burke R. Hybrid recommender systems: survey and experiments. User Model User-Adap Inter. 2002;12:331–70.
87. Gunes I, Kaleli C, Bilge A, Polat H. Shilling attacks against recommender systems: a comprehensive survey. Artif Intell Rev. 2012;42:767–99.
88. Park DH, Kim HK, Choi IY, Kim JK. A literature review and classification of recommender systems research. Expert Syst Appl. 2012;39:10059–72.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.