



Efficient Machine Learning Algorithms in Hybrid Filtering-Based Recommendation Systems

Ruchika * 

*Corresponding author, Assistant Professor, Amity Institute of Information Technology, Amity University Uttar Pradesh, India. E-mail: bathla.ruchika@gmail.com

Mayank Sharma 

Assistant Professor, Amity Institute of Information Technology, Amity University Uttar Pradesh, India. E-mail: msharma22@amity.edu

Syed Akhter Hossain 

Ph.D., Dean, School of Science and Engineering, Canadian University of Bangladesh, Dhaka, 1212, Bangladesh. E-mail: akhter.hossain@cub.edu.bd

Abstract

The widespread use of E-commerce websites has drastically increased the need for automatic recommendation systems with machine learning. In recent years, many ML-based recommenders and analyzers have been built; however, their scope is limited to using a single filtering technique and processing with clustering-based predictions. This paper aims to provide a systematic year-wise survey and evolution of these existing recommenders and analyzers in specific deep learning-based hybrid filtering categories using movie datasets. They are compared to others based on their problem analysis, learning factors, data sets, performance, and limitations. Most contributions are found with collaborative filtering using user or item similarity and deep learning for the IMDB datasets. In this direction, this paper introduces a new and efficient Hybrid Filtering-based Recommendation System using Deep Learning (HFRS-DL), which includes multiple layers and stages to provide a better solution for generating recommendations.

Keywords: Recommender System; Content-Based Filtering; Collaborative Filtering; Movie Recommendation; Deep Learning.



Introduction

E-commerce and marketing companies use data and improve sales through promotional systems on their websites and programs. The application of Recommender Systems (RS) has been increasing steadily in recent years. They have been instrumental in E-commerce, improving customer experience, product promotion, and product ratings. It eliminates the tyranny of choices, smoothing the way for decision-making and increasing online sales. Nowadays, the use and applications of RS are taking their pace with various Machine Learning (ML) techniques.

Machine learning is a branch of computer science where we learn about computer algorithms that automatically improve with the help of experience and data usage. Artificial intelligence (AI) includes machine learning as a subset. It uses training data and modelling to generate predictions and judgments. The applications of ML have taken a giant form and have widespread use cases in various fields like sentiment analysis, generating recommendations, email filtering, and image processing, etc. Deep learning is a subfield of ML from various data abstractions and representation levels. Many industries, corporations, and companies already use Deep Learning-based RS (DLRS) built upon different neural networks to improve customer experience. For Example, YouTube, Netflix, eBay, Twitter, etc., choose deep neural networks, while apps like Spotify use a Convolutional Neural Network (CNN). Deep learning-based recommender systems cope with complex interaction patterns and precisely reflect users' preferences. Given effective feature extraction, CNN is a good fit for unstructured multimedia data processing. Also, CNN helps us remove the cold start problem and overcome the drawbacks of traditional systems like collaborative filtering.

The DLRS helps the users get personal recommendations to make correct decisions related to business needs or individual requirements, including online transactions, sales redefining users' web browsing experience, and improving their shopping experience. It can change how the website communicates with the users, which can help them improve their ROI (Return on Investment). It also makes an organization more customer-centered as all the information gathered is based on customer requirements, like what customers prefer or purchase. In this way, it is beneficial for both service providers and users. It predicts whether a particular customer would choose an item or not; hence, it helps them save time. As a result, it makes product recommendations to clients based on their tastes, assisting them in selecting the best product and enhancing customer experience. In addition, it provides the personalization of products and promotes one-to-one marketing. For example, Amazon, one of the biggest e-commerce websites, uses RS and provides its users with the best choice, which improves customer experience and effect. This is just one example from the e-commerce industry; many other service providers also use RS. Another example is a music application that uses a recommender system to suggest songs based on your preferences, giving users many choices to select and have a great experience.

The following is how this paper is put together. The first section gives a general overview of recommendation systems and a taxonomy of RS. The subsequent section compares four surveys of existing RSs, their various parameters, performance metrics, learning paradigms, techniques, challenges, limitations, and needs. The next section demonstrates the difficulties and problems encountered in existing RSs. The following section presents a graphical analysis of various parameters of existing algorithms. The proposed methodology is discussed in the last section, followed by the conclusion.

Literature Review

In today's era, the widespread and frequent use of the E-commerce industry requires the hybridization of prominent fields such as NLP, ML, and text mining. The E-commerce industry encompasses a diverse set of websites that sell and buy many products and services worldwide. Such websites face competition and need customer feedback, reviews, and suggestions for each perspective. In addition, these online websites suggest similar products, services, or offers to their users or customers using content, collaborative or hybrid filtering techniques. A Movie Recommendation System (MRS) is a system that uses users' prior experience as input and makes predictions. Netflix is a platform that houses thousands of movies and television shows in one location, and users can watch any movie anytime. It considers the user's previous watch history and liked movies and then recommends more movies based on their interest.

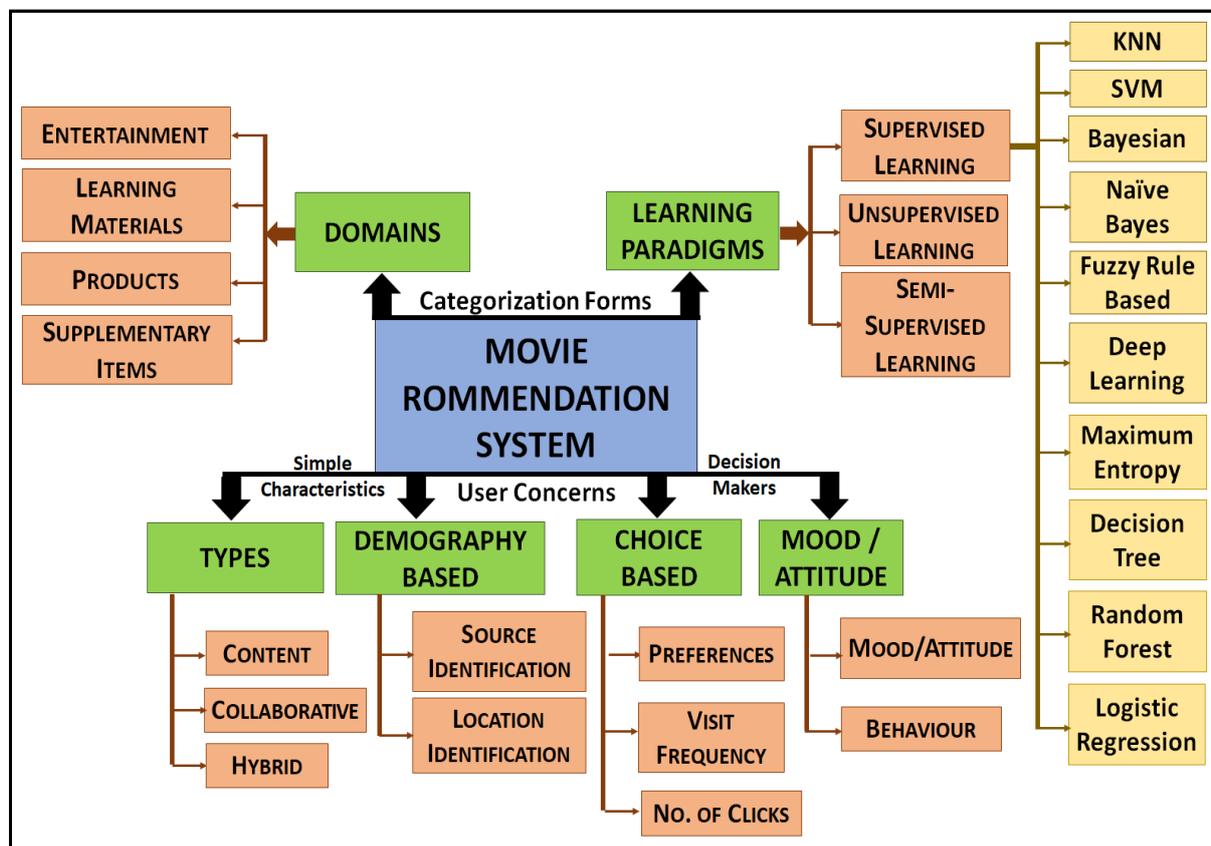
The MRS defines categorisation-based and user-based issues at the primary level for recommendations. The categorization form consists of domains and learning paradigms, as shown in Figure 1. There are four domains: entertainment, learning materials, products, and supplementary items. Unknown datasets are classified into established categories using the supervised learning technique, which uses a known dataset to train. The unsupervised learning method is the inverse of the supervised learning method in categorizing the comments without prior knowledge. The Semi-supervised learning technique is the combination of supervised as well as unsupervised learning techniques.

The Support Vector Machine (SVM) is a technique used for dealing with binary classification problems. This linear classifier is expanded to address situations involving multiple classifications. The fuzzy rule-based systems use the uncertainty and membership concepts of the fuzzy sets and fuzzy logic. They show different types of knowledge, how the interactions work, and how the variables are linked together. Another supervised learning method, Artificial Neural Networks (ANN), comprises connected input-output networks. The weights are assigned to the connections and updated if an error is discovered. It generally consists of an input layer with multiple hidden layers and an output layer. The Maximum Entropy (ME) classifier is a probabilistic classifier that belongs to the exponential model category. A decision tree is similar to a flow chart in principle. It has a tree-like structure and

categorizes instances based on their characteristics. The supervised learning approach Naive Bayes (NB) discovers substantial independence among the features discovered. Logistic Regression (LR) is a supervised classification technique that uses the probability value to determine the classification. SentiWordNet is an opinion lexical analysis resource derived from the WordNet database. A Random Forest (RF) is a group of trees that are different from each other. It is a supervised learning algorithm used for regression and classification problems. The K-Nearest Neighbor (KNN) method is used for classification and regression. It reserves all the cases available during training and then ranks the new instances using many votes of its K neighbors.

Figure 1.

Taxonomy Showing MRS-based Factors



Decision-based methods are influenced by the user's behavior and mood/attitude. Content, collaborative, and hybrid filtering can provide a solution for MRS. The source-based techniques focus on the data source and location identification. Decision-making strategies such as behavioral and mood-based techniques solely depend on the user. The mood or attitude also contributes to determining the polarity and orientation of the comments. The choice-based concern includes the user preferences, their visit frequency to the website, and the number of clicks on a particular website or product.

Role of Collaborative Filtering in AI and Deep Learning

It's important to note that the recommendation systems are classified into three categories: Collaborative (CF), Content-based Filtering, and hybrid Filtering. The user's item and profile features are primarily used to develop content-based recommendations, whereas collaborative filtering uses similar audience preferences. The collaborative filtering Strategy explores the concept of the relationship that exists between people's interests and products. Many recommendation systems use hybrid filtering to identify these relationships and provide accurate product recommendations of customer interest. It is a standard method for developing automated recommendation systems to give better recommendations as more user information is gathered. This way, one can filter the user's interests based on the same type of users' reactions, comments, and responses. It works by searching for many people and finding a small set of users with similar interests. In such a way, they put them together to create a ranked list of suggestions.

CF is based on the relationships and choices made by the users when they purchase something from the Shopping website. It enables companies to connect users with similar interests by generating predictions. An example of collaborative learning is Netflix; everything we see on the site is chosen by customers who are made often enough to turn into recommendations. The Netflix app directs the recommendation so that the product at the top is more visible to users; hopefully, they will also choose the recommendations. Another example is the shopping website Amazon. The recommendation system is based on previous purchases, the quantity you have ordered, and other factors learned from earlier visits to their websites. The benefits of CF in AI and in-depth learning give broader exposure to a variety of products in which they may be interested. This exposure provides users with a continuous process for purchasing a product. It also provides support to the service provider and provides a better experience. Collaborative filtering is classified as follows:

User-User Based Collaborative Filtering: The idea behind this filtering is pretty simple. Based on the product rating history, it finds other users who look similar to a particular user. It is also determined whether or not their preferences and recommendations for the products are the same. For instance, two people are there, i.e., Lennie and Bob both liked the first-star movie and watched the next star movie. So, it seems that to recommend empires strike to Bob, and if bob loves that and Lennie loves the movie, there will be an excellent chance that they both like each other movie preferences. Here, the steps include taking the sample data with the ratings for everyone in the system. Consider a two-dimensional array with the movie on one axis, users on the other, and ratings in each cell. There will be a five-dimensional vector if there are five movies, so we must compute the cosine similarity score between any two users. One can define it with similar metrics to the other two. So, all these will fall into the algorithm and makes intuitive sense looking at the results. Sparse data is a massive problem with collaborative filtering in general, and it can lead to weird results, so we need to

take a minimum threshold value for each. We can give each person a score; the last step is filtering out.

Item-based Collaborative Filtering: Item-based collaborative Filtering finds similarities between all two objects. First, the system uses a model-building phase. This similarity function will take many forms; the relationship between the measurements and the cosine of those measurement carriers is explained with the help of this. In user-user systems, the same functions can use the standard parameters, e.g., adjustment, with each user rate from time to time. Second, the program creates a recommendation phase, producing a list of recommendations for using the most similar items in user properties already rated. Usually, this calculation is based on a weighted scale. It has a much smaller error than the user-user filtering technique and has a less-dynamic model.

A review tour of existing Recommendation Systems

The literature survey on existing recommender systems uncovered a plethora of contributions. This section covers the years 2012 to 2020. It discusses the four surveys in detail in each of the four sections. These algorithms are examined on various criteria, and their drawbacks and limits are discussed and contrasted.

Deep Learning-based Hybrid using Movie Datasets (2015-2020)

This section illustrates the existing RSs evolved from 2015 to 2020. They depict deep learning-based hybrid filtering using movie datasets. The hybrid filtering-based Recommendation System (RS) (Wang et al., 2015) used advanced deep learning techniques, i. e., the hierarchical Bayesian model, from I. I. D. input to non-I. I. D. input. It combined learning for the content information and collaborative filtering to get the rating matrix. The system's total time complexity was $O(K_1 + K_2 J_2 + K_2 I_2 + K_3)$. Another reliable recommendation method (Subramaniaswamy et al., 2017) suggested items based on the user's interests through clustering and filtering techniques. This method helped the system understand the user and develop suggestions by obtaining the movie ratings from the users. So, its architecture followed the steps of data acquisition and repository, recommendation, using the user interface.

Another deep learning-based recommender (Zhang et al., 2018) performed feature set extraction using quadratic polynomial regression and obtained the latent features by improving the matrix factorisation method. The other deep neural networks used these features to predict the rating scores and obtained results in terms of Mean Absolute Error (MAE). This algorithm (Zhang et al., 2018) proposed compares the Singular Vector Decomposition (SVD), Probabilistic Matrix Factorization (PMF), item based, MCoC, DsRec, Proximal Support Vector Machine (PSVM), Self-Constructing Clustering (SCC), PMMMF, and TyCo for feature reduction. (Zhang et al., 2019) provided the review of the state-of-the-art

and recent research contributions on deep learning-based recommender systems. Another state-of-the-art survey (Shokeen & Rana, 2020) described the recommendation systems' existing approaches, domains, parameters, metrics, datasets, and future perspectives. The subsequent state-of-the-art study (Goyani & Chaurasiya, 2020) included the recent movie recommenders' reviews, limitations, gaps and challenges. Table 1 compares the problems focused on existing algorithms and their filtering type, data source, domain, data size, data type, and other related features and parameters. Table 2 discriminates the performance metrics, learning techniques, learning type, challenges, limitations, and future enhancements needed.

Table 1.

Comparing Problem Analysis, Filtering Type, Data Source and Domain, Data Size and Data Type, and Other Features of Existing Approaches - Survey 1

Reference	Problem Analysed	Filtering Type	Data Source & Domain	Data Size & Data Type	Other Features & Parameters
(Wang et al., 2015)	Design of recommendation system using the source of information.	Hybrid	CiteULike3 & Netflix. 3 Datasets = 2 (CiteULike3) + 1 (Netflix).	Three hundred seventy-three samples (298 training + 76 testings). Seed tags, articles & movies.	The dense parameter $p = 1, 10$ etc.
(Subramaniaswamy et al., 2017)	Personalised recommendation system design for movies.	Collaborative	Research Project Website. MovieLens,	100 K Samples, eight users. One million Anonymous Movie Ratings.	Recommended the movies based on the nearest neighbor's best-rated movies.
(Zhang et al., 2018)	Movie recommender builds user images via the user's rating information, features, CNN, cluster & recommendations.	Content & Collaborative	Epinions & MovieLens. 100 K MovieLens & 1 M Epinions.	100 K Samples (80 K Training, 20 K Testing). Movie Ratings.	$a = 16$ & $b = 18$ for MovieLens-1K; $a = 20$ & $b = 20$ for MovieLens-1M; for Epinions. Learning rate $\eta = 0.01$.
(Zhang et al., 2019)	Survey with new directions on movie recommendations.	Content, Collaborative & Hybrid	Internet Movie Database (IMDB). MovieLens.	Sufficient data set of movies.	-
(Shokeen & Rana, 2020)	Designing a Social Recommendation system	Collaborative	Netflix. MovieLens.	Sufficient data set of movies.	Worked on the taxonomy of recommenders
(Goyani & Chaurasiya, 2020)	Review of Existing Movie Recommendation Systems	Content, Collaborative & Hybrid	Netflix. MovieLens.	Sufficient data set of movies.	Evaluated many similarity measures.

Table 2.

Discriminating Performance Metrics, Learning Techniques, Learning Type, Challenges, Limitations, and Needs of Existing Approaches - Survey 1

Reference	Performance Metrics	Classification/ Clustering/ Prediction & Type Name	Limitations, Threats, and Needs
(Wang et al., 2015)	Best Recall: 59.43% (citeulike-a), 54.48% (citeulike-t) & 70.42% (Netflix).	Prediction. Deep learning (Bayesian Model).	Need to extend with other models also.
(Subramaniaswamy et al., 2017)	Recall=95.1% & Precision=96.1%.	Classification. CF with two use cases.	The challenge is to change the movie preferences with time—extension with demographic information for better recommendations. Extend with mobile apps & other real-life user interests.
(Zhang et al., 2018)	Promising Results with MAE.	Prediction. Deep Neural Networks.	Algorithmic performance is limited to the high sparsity of data. Enhancements needed: To build highly complex systems & use other deep learning methods. Non-ideal performance of SVD.
(Zhang et al., 2019)	Good Results.	Prediction. Deep Learning.	Improvements in performance parameters must be worked on.
(Shokeen & Rana, 2020)	Promising Results.	Prediction. Deep Learning.	Efforts must be made to improve performance parameters.
(Goyani & Chaurasiya, 2020)	Promising Results.	Prediction. Deep Learning.	Need to use recommenders to increase profit and for the benefit of customers.

Deep Learning-based Hybrid using Random Datasets (2015-2018)

This section illustrates how the existing RSs evolved from 2015 to 2018. They depict deep learning-based hybrid filtering using random datasets. Another recommender system (Lu & Zhang, 2015) used the tree-structured design to frame author features, such as biography, written book introduction & comments, & tri-layered MLSOM to handle the authors. It went through pre-processing, word extraction, vocabulary building, and making the PCA projection matrix. The prediction approach (Lin, 2017) used deep learning and semantic Chinese TCM telemedicine system with ANN. It included four processing steps: questioning/history data; inspection; auscultation (listening) and olfaction (smelling); and palpation. Another recommender (Wang et al., 2018a) proposed Deep Knowledge-Aware Network (DKN) method with click-through rate prediction for highly time-sensitive news. Here, the word-entity-aligned Knowledge-aware CNN (KCNN) combines both semantic and knowledge levels, whereas the attention module dynamically aggregates the user's history and the latest candidate news. The results of (Wang et al., 2018a) outperformed baselines by 2.8% to 17.0% on F1 and 2.6% to 16.1% on AUC, where the significance level was 0.1. Table 3 compares the problems focused on existing algorithms and their filtering type, data source, domain, data size, data type, and other related features and parameters. Table 4 discriminates the

performance metrics, learning techniques, learning type, challenges, limitations, and future enhancements.

Table 3.

Comparing Problem Analysis, Filtering Type, Data Source and Domain, Data Size and Data Type, and Other Features of Existing Approaches - Survey 2

Reference	Problem Analysed	Filtering Type	Data Source & Domain	Data Size & Data Type	Other Features & Parameters
(Lu & Zhang, 2015)	Multilayer SOM based Recommendation System using Tree Structure	Content	Amazon. Books.	7426 Authors (6684 Training, 742 Testing). 205805 books & 3027502 comments.	C = 0.5 to 0.8 & Pool Size = 11
(Lin, 2017)	Deep learning Application and Analysis for Recommendation.	Content	Microsoft Azure. Telemedicine.	100 Clinical Training Cases. Cough-based.	Used filter-based feature selection.
(Wang et al., 2018a)	Knowledge-Aware Network using Deep Learning	Content	Bing News. News Articles.	Random balanced: October 16, 2016, to June 11, 2017 (Training) & June 12, 2017, to August 11, 2017 (Test).	Confidence > 0.8. Set dimensions of word & entity embeddings, filter window sizes & several filters.

Table 4.

Discriminating Performance Metrics, Learning Techniques, Learning Type, Challenges, Limitations and Needs of Existing Approaches - Survey 2

Reference	Performance Metrics	Classification / Clustering / Prediction & Type Name	Limitations, Threats, and Needs
(Lu & Zhang, 2015)	Results > 80%	Prediction & Clustering. Multilayer SOM	Need to get a more effective recommender for e-book authors. Extend for more MSOM applications.
(Lin, 2017)	Accuracy=77%	Prediction. Deep Learning.	It is necessary to deploy as a SaaS and Integrative Medicine Model. Extend the GPU Visualization Infrastructure and GPU Compute Infrastructure with Microsoft Azure and NVidia.
(Wang et al., 2018a)	F1: 68.9 ± 1.5, AUC: 65.9 ± 1.2	Prediction. Deep Knowledge-Aware Network	Good knowledge & attention module usage with 3.5% and 1.4% improvement.

Hybrid using Other Techniques and Movie Datasets (2013-2020)

This section illustrates the existing RSs evolved from 2013 to 2020. They depict the hybrid filtering using other techniques and movie datasets. The new recommendations were created by combining media ratings and unrated user comments with sentiment knowledge included in recommendations in the TV media recommender system (Peleja et al., 2013). This approach improved the popularity of specific entertainment programs and shows. It performed matrix factorisation by Singular Value Decomposition (SVD) to evaluate explicit ratings and sentiment analysis results. The method in (Liu et al., 2014) used Users' Tastes and Users' Choices to promote the movies. It included the layered architecture to show their relationships

and use the "Maslow's Hierarchy of Needs theory". A hybrid model-based intelligent movie recommender (Wang et al., 2014) proposed a cluster-based CF method, i.e., optimized K-means clustering coupled with genetic algorithms (GA-KM) to partition the transformed user space. Its offline phase trained the low-dimensional clustering model to target active users. In contrast, the online stage prepared the TOP-N movie recommendation list using historical rating data for active users. It uses principal component analysis (PCA) to condense the space occupied by movie populations, handling high dimensionality and data sparsity issues.

The survey (Nagarnaik & Thomas, 2015) included many recommendation-based techniques and presented a web page recommender using Collaborative Filtering, CHARM algorithm, clustering, and association rule mining. The review (Harper & Konstan, 2015) on the MovieLens Datasets covered its historical perspectives and discussed the findings from running a research organization's long-standing, live research platform. According to the recommendation systems survey (Chen et al., 2015), two current state-of-the-art studies use reviews to build user profiles and product profiles. The next recommender system (Aggarwal, 2016) used an ensemble approach and hybrid techniques to enhance the performance of existing systems using specific data modalities. It predicted and validated the data using a content-based approach using collaborative information as features.

The recommendation method proposed (Salam & Najafi, 2016) compared the accuracy of small and large datasets algorithms using the matrix factorization algorithm called FunkSVD. Another approach (Huanyu et al., 2016) first selected the similar type of users and then calculated the users' similarities. Then it used the user-item bipartite graph with the shortest path algorithm to locate the candidate items. Finally, it rated the data using graphs. The following method (Christakopoulou & Karypis, 2016) obtained the prediction scores as a user-specific combination using the global and local item-item models. It acquired the Top N-recommendations through SLIM (Sparse Linear Methods) in a personalized way automatically. (Katarya & Verma, 2016) presented the movie recommendation system using the type of division method, which classified the movies based on users and reduced the complexity. It used K-Means and Fuzzy C-means methods to get initial parameters and improve performance.

The online recommender system (Gurcan & Biturk, 2016) used content-boosted collaborative filtering with dynamic fuzzy clustering (CBCFdfc) to solve sparsity, new item and over-specialisation issues. The fuzzy clustering-based CBCFdfc increased prediction accuracy but decreased the online prediction time. Here naive Bayesian (NB) outperformed Melville et al.'s method using the average likelihood decision rule and adjustment value. It computed the final predictions with multiple clusters and evaluated the Mean Absolute Error (MAE) and Receiver Operating Characteristic (ROC) results. The recommendation system (Saipraba & Subramaniaswamy, 2016) improved the stability and accuracy using ensemble-based techniques. It first divided the data into a ratio of 80:20 for the training & testing stages,

respectively, and then evaluated their stability. After that, training data is further divided into the proportion of 75:25, and the 75% available ratings are used to apply the boosting, bagging, or smoothing technique. Then it selected an approach from the set, say user-based CF, item-based CF, Bayesian Probabilistic Matrix factorisation and SVD. Then the 25% training data was used to evaluate the accuracy (RMSE) and computed and validated the stability (RMSS) by making predictions suggest recommendations to active users. The recommender system (Sattar et al., 2017) used a hybrid filtering framework with fivefold cross-validation on training data. It followed the steps of data acquisition, pre-processing, feature selection, searching neighbours of unknown items, obtaining these neighbours' crawling information, and prediction.

The cross-domain recommendation system (Subramaniaswamy & Logesh, 2017) included the knowledge-based domain-specific ontology model to generate personalised recommendations. It first used two mini and prominent representation models to create a good set of suggestions. It then predicted the data by correlating the user preferences and item features. Another system (Katarya & Verma, 2017) predicted the movie data through data clustering and computational intelligence. It evaluated the metrics such as standard deviation (SD), MAE, root mean square error (RMSE) and t-value to get better results. The review (Wasid & Ali, 2017) surveyed various recommendation systems related to soft computing techniques. It also discussed the future scope with FS, NN, EC, and SI methods. Another movie recommender (Wang et al., 2018b) first created the preliminary recommendation list and then optimized it using sentiment analysis, and lastly, this analysis was implemented on the Spark platform. The following review (Portugal et al., 2018) illustrated the use of ML techniques and the scope of the recommendation techniques in software engineering research.

The FP-Tree-based movie recommendation system (Tuan et al., 2018) evaluated the users' ratings and behaviors to suggest the most suitable and desired movies to the active users. It went through pre-processing, building FP-tree, & recommendation engine. The next unified recommendation system (Katarya, 2018) predicted the data using Artificial Bee Colony-K-Means (ABC-KM) as an optimization procedure and improved the recommender systems. (Jain & Gupta, 2018) qualitatively and quantitatively analyzed the growth of fuzzy logic in recommendation systems and their application areas. The movie recommendation system (Sadanand et al., 2018) uses a hybrid algorithm and Apache Spark to implement the user-user similarity, item-item, Tanimoto, Pearson coefficient, Slope one, and SVD recommendation algorithms. An efficient recommendation system (Vilakone et al., 2018) analyzed the social networks using the k-clique method and compared the MAE results of improved k-clique, k-clique, KNN-CF, and Maximum clique algorithms. Another analysis (Menon & Paul, 2019) evaluated and designed the recommendation system using a simulated annealing-based K-Means clustering model.

The recommendation system (Indira & Kavithadevi, 2019) applied a multi-cloud environment and ML to enhance ranking and search quality. It improved the speed of the work and provided better-prioritised user output. It followed the pre-processing stages, feature selection using PCA, clustering using the Hierarchical Agglomerative Clustering algorithm (HAC) and k-means, cluster ranking using the trust ranking algorithm, and evaluation and analysis of performance measurements. Another optimised algorithm (Selvi & Sivasankar, 2019) applied a modified fuzzy c-means clustering (MFCM) approach to get a cluster with reduced errors; used the MFCM clustering approach for validation; obtained optimal users in each cluster using MCS techniques tested and evaluated the MCS. The next system (Gunjal et al., 2020) applied the offline phase for SVD-based dimension reduction and made the clusters of the most similar users and items. The online step used incremental SVD to find the most relevant item. It used the MAE and Root Mean Squared Error (RMSE) performance parameters. Table 5 compares the problems focused on existing algorithms and their filtering type, data source, domain, data size, data type, and other related features and parameters. Table 6 discriminates the performance metrics, learning techniques, learning type, challenges, limitations, and future enhancements needed.

Table 5.

Comparing Problem Analysis, Filtering Type, Data Source and Domain, Data Size and Data Type, and Other Features of Existing Approaches - Survey 3

Reference	Problem Analysed	Filtering Type	Data Source & Domain	Data Size & Data Type	Other Features & Parameters
(Peleja et al., 2013)	TV media Recommendation System with a combination of movie ratings & unrated reviews.	Content & Collaborative	IMDB, TSA09, Amazon & other Video OnDemand (VoD) Services.	2000 Samples (1400 Training, 6000 Testing). Dataset: Polarity (Sentiment Analysis), Movies & Music.	Used SVD. It can be used to filter spam.
(Liu et al., 2014)	Multilayer Recommendation to Promote the Movies.	Collaborative	IMDB, MovieLens-100K & MovieLens-1M.	100K Samples in a 4:1 ratio.	Handled the Users' Tastes & Choices.
(Wang et al., 2014)	Computational intelligence-based improved movie recommendation.	Collaborative	IMDB, MovieLens.	Nine hundred forty-three users on 1682 movies. Movies	Used PCA. Sparsity = 0.9369, Cluster Number K = 16. Used "like-minded" neighbourhoods to get common ratings in high-quality recommendations.
(Nagarnaik & Thomas, 2015)	Review on Recommendation Systems	Hybrid Collaborative	IMDB, MovieLens.	Movies	Web Page Recommendation.
(Harper & Konstan, 2015)	Discussion on Historical Aspects of MovieLens Datasets	Collaborative	IMDB, MovieLens.	Sufficient Data Sets. Movie Ratings.	Discussed many features of the recommender systems.
(Chen et al., 2015)	Survey on Recommendation Systems based on	Content & Collaborative	IMDB, MovieLens.	Sufficient Data Sets. Movie Ratings.	Used user & product-based Profiles. Rating

	User Reviews.				enhanced using term-based profiles of user profiles.
(Aggarwal, 2016)	Hybrid Recommendation System using ensemble Approach	Content cum Collaborative	Multiple Data sources	Sufficient Data Sets.	Worked on features of multiple data modalities.
(Salam & Najafi, 2016)	Evaluation of Prediction Accuracies of Recommenders.	Collaborative	IMDB. MovieLens.	943 Users On 1682 Movies. 100.000 Ratings.	Used small & large datasets.
(Huanyu et al., 2016)	Recommender with improved Collaborative filtering.	Collaborative	IMDB. MovieLens.	Movies	Improved Collaborative.
(Christakopoulou & Karypis, 2016)	Improved Top-N Recommender using Local Item-Item Models.	Content	IMDB. MovieLens. Groceries, ML, Jester & Flixster.	Items, Jokes & Movie Ratings & 3500 Ratings.	The global and local models, their user-specific combinations, and the assignment of users to local models were all jointly optimised.
(Katarya & Verma, 2016)	Hybrid movie recommender system to improve movie prediction, accuracy & user recommendation.	Collaborative	Kaggle. MovieLens.	943 Samples (500 Training & 443 Testing). Movie Ratings.	Used optimisation algorithm. Obtained 3.503 % better results with MAE = 0.75. Used Fuzzy C-Mean & K-Mean Clustering.
(Gurcan & Birturk, 2016)	Hybrid recommendation approach for better classification of movies.	Content boosted Collaborative	IMDB. MovieLens.	10 Samples (7-Training, 3-Testing). Hybrid Movies.	Set values of AV, NS, ST, CW & NAC parameters. Used dynamic Fuzzy clustering & a user interface to check user opinions.
(Saipraba & Subramaniaswamy, 2016)	Ensemble-based Information Retrieval to improve recommender's stability	Collaborative	IMDB. MovieLens.	943 Users, 100k Records With 100,000 Ratings.	Focused on stability features. Increased stability using Boosting, Bagging & Smoothing.
(Sattar et al., 2017)	Automatic recommender using ML	Hybrid Filtering	IMDB. MovieLens (SML) & FilmTrust (FT).	Rating Datasets: 943 Users, 1682 Movies & 100K Ratings (SML) & 1592 Users, 1930 Movies & 28645 Ratings (FT).	Parameter K=13. Used feature extraction.
(Subramaniaswamy & Logesh, 2017)	Personalised recommendation system for online items.	Collaborative	IMDB. MovieLens.	943 Users, 1682 Movies & 100K Ratings.	Used domain-specific ontology. Item clustering used similarity features.
(Katarya & Verma, 2017)	Recommendation approach to classifying users of similar interests.	Collaborative	IMDB. MovieLens.	943 Users, 1682 Movies & 100K Ratings with a Scale of 1-5.	Used many parameters.
(Wasid & Ali, 2017)	Review on recommenders using Soft	Collaborative	IMDB. MovieLens.	Movie Ratings	-

	Computing.				
(Wang et al., 2018b)	Sentiment-Enhanced Hybrid Movie Recommender	Content, Collaborative & Hybrid.	IMDB. MovieLens.	Movie Ratings	Fast method.
(Portugal et al., 2018)	Review on Recommenders using ML	-	IMDB. MovieLens.	Movie Ratings.	Most of the contributions used Bayesian and decision tree algorithms.
(Tuan et al., 2018)	FP-tree & Movie Recommendation based on user ratings & behaviours.	Content & Collaborative	IMDB. MovieLens.	20 M Movie data sets.	Satisfaction m=5. Used level of rating criteria.
(Katarya, 2018)	Recommender to reduce the cold start problem.	Content	IMDB. MovieLens.	100K Ratings, 943 Users & 1682 Movies.	Used scalability to achieve a great level of performance.
(Jain & Gupta, 2018)	Year-wise analysis of recommendation systems and growth of Fuzzy-based systems.	Content	Different Data sets.	Analysis from the year 2003 to 2016.	Analysed the features carefully for optimal prediction because of imprecise, uncertainty & vague user profile.
(Sadanand et al., 2018)	Automatic Movie Recommender Engine	Content, Collaborative & Hybrid.	IMDB. MovieLens.	Seventy-one thousand five hundred sixty-seven users on 10681 movies.	Use of unstructured & semi-structured data. Smart clustering is achieved through hybrid algorithms.
(Vilakone et al., 2018)	Recommendation System to analyse social networks	Collaborative	IMDB. MovieLens.	# Of Users: 800 in experimental & 143 in test data. 100K ratings from 943 users on 1684 movies.	Parameter K = 3 to 14. Can analyse social networks.
(Menon & Paul, 2019)	Designed the clustering process for recommendations.	Content & Collaborative	IMDB. MovieLens.	590 Movie Samples (472 Training, 118 Testing) & Tags.	Can solve the local minima problem—item description through keywords.
(Indira & Kavithadevi, 2019)	Fast Recommender using ML & Multi-cloud	Content	IMDB. MovieLens.	705309 user reviews, 1-5 ratings & 522 movies.	Feature selection through PCA, K-means & HAC.
(Selvi & Sivasankar, 2019)	Efficient Recommendation system using optimised algorithms.	Collaborative	IMDB. MovieLens.	100K Ratings, 1000 Users & 1700 Movies.	PSO and CS converged with fewer iterations and a lower minimum fitness value.
(Gunjal et al., 2020)	Hybrid scalable recommendation system using ontology and incremental SVD.	Collaborative	IMDB. MovieLens. & Flixter.	786936 Users, 48794 Items & 8196095 Ratings.	Handled data sparsity, scalability, & significant prediction errors.

Table 6.

Discriminating Performance Metrics, Learning Techniques, Learning Type, Challenges, Limitations and Needs of Existing Approaches - Survey 3

Reference	Performance Metrics	Classification / Clustering / Prediction & Type Name	Limitations, Threats, and Needs
(Peleja et al., 2013)	Good Results	Classification. Sentiment-based recommendation.	Performance issues with 1-star and 5-star ratings. Extension into integrated Web TC with view & comment parameters. Need to improve the prediction ratings by identifying spam reviews.
(Liu et al., 2014)	Got better results than existing systems.	User-User Similarity.	Need to extend with other better layering approaches.
(Wang et al., 2014)	Promising Results	Prediction. GA-KM.	Need improvements. Need to handle issues of high dimensions & sparsity. Extend highly personalized recommenders with tags, context & web of trust.
(Nagarnaik & Thomas, 2015)	Good Results	Clustering. K-Means.	Need to extend to many other approaches.
(Harper & Konstan, 2015)	Good Results	Prediction Analysis.	It included the limitations of the MovieLens Dataset.
(Chen et al., 2015)	Good Results	Prediction Analysis.	Need to extend to many other approaches.
(Aggarwal, 2016)	Good Results	Prediction. Content-based Algorithm.	Extend using a greater number of hybrid recommender systems.
(Salam & Najafi, 2016)	Promising Results; Better accuracy than existing ones	Prediction. SVD.	We need to extend with other properties of testing methods.
(Huanyu et al., 2016)	Promising Results	Parallel Graph. Improved Similarity Model.	Need to extend with other approaches.
(Christakopoulou & Karypis, 2016)	GLSLIM had 17% improved results than Top-N recommenders.	Prediction. GLSLIM (Global and Local SLIM).	Need to improve the results more & extend with the other approaches.
(Katarya & Verma, 2016)	0.78 MAE	Classification. Particle Swarm Optimisation.	We need to include the features, say age, etc., to get more reliable & accurate rating results.
(Gurcan & Birturk, 2016)	Accuracy= 82% to 85%. Better accuracy results with CBCFdfc than CBCFonl	Classification with Naive Bayesian. Fuzzy Clustering.	Need to extend with item-based similarity—the results dropped from O (n) to (1) regarding online recommendation time.
(Saipraba & Subramaniaswamy, 2016)	Accuracy: Boosting= 2.1 to 2.5, & Other algorithms= 1.55 to 1.92.	Prediction. Bayesian Probabilistic Matrix.	Need to work more on the stability factor. Need to enhance the system for biased (feel-based) & unbiased (purpose-based).
(Sattar et al., 2017)	Promising Results	Prediction. NB, Bayesian classifier, SVM, Decision tree over K-selected neighbours.	It is necessary to increase the value of parameter C to create an accurate model.
(Subramaniaswamy & Logesh, 2017)	93.87% Accuracy	Prediction. Adaptive KNN.	Sparsity, scaling, and cold-start issues must be dealt with. Of all algorithms, CPAR had the worst results.
(Katarya & Verma, 2017)	MAE = 68% to 80%. 0.68 MAE with K-mean Cuckoo resulted in better than 0.78 MAE of existing work & 0.75 MAE of our previous work.	Classification, K-Means by Cuckoo Search.	The initial partition did not work well—which decreased efficiency.
(Tuan et al., 2018)	Accuracy (Precision & Satisfaction): 96% with m = 5 & 98% with 5-fold cross-validation.	Clustering. Frequent-Pattern Tree (FP-Tree)	Need to increase the efficiency more.
(Katarya, 2018)	Better precision than existing ones, say	Meta-Heuristic Artificial Bee Colony & K-Means.	Need to improve the MAE, precision, recall, and accuracy results.

	PCA, GA-KM & SOM.		
(Jain & Gupta, 2018)	Promising Results	Prediction. Fuzzy Logic.	Issues found: Lack of handling data with imprecise information & gradualness.
(Sadanand et al., 2018)	Promising Results	Prediction & Clustering. Tanimoto, Pearsons, Slope & SVD Algorithms.	Need to increase the efficiency more.
(Vilakone et al., 2018)	Precision= 61%. Improved K-Clique provides the best precision	Prediction. K-Clique, KNN & Maximal Clique.	Need to use data mining to increase accuracy.
(Menon & Paul, 2019)	Promising Results	Improved Clustering. Simulated Annealing in K-Means.	Need to increase the efficiency more.
(Indira & Kavithadevi, 2019)	High Recall Rate & High Precision	Prediction. Trust Ranking Algorithm.	Need to increase the efficiency more.
(Selvi & Sivasankar, 2019)	Precision = 71%-75%, F-measure = 83%-86% & Recall = 100%	Clustering. MFCM Clustering & Optimised Cuckoo Search (MCS) Algorithm.	Need to increase the efficiency more.
(Gunjal et al., 2020)	RMSE = 81%-84% & MAE = 61%-63%	Clustering. Ontology & incremental SVD.	Need to improve accuracy prediction using KNN & ontology approaches.

Hybrid filtering using Other Techniques and Random Datasets (2015-2020)

This section illustrates how the existing RSs evolved from 2015 to 2020. They depict hybrid filtering using other techniques and random datasets. The Personalised RS (Ojokoh et al., 2012) intelligently predicted the information of the product features and suggested the optimal professional services and products to the active users using near fuzzy compactness. The system was designed to improve sales for online businesses. The simulator (Kowalczyk et al., 2011) checked the diversity impacts of the movie RS. It included the number of scenarios, simulator runs, diversity, and validation-based analysis, & the report on selected observations. The Higher-Order Sparse Linear Method for Top-N Recommender Systems (HOSLIM) approach (Christakopoulou & Karypis, 2014) learned two sparse aggregation coefficient matrices, S and S0, to capture item-item and itemset-item similarities, respectively. The RS (Gogna & Majumdar, 2015) applied the low-rank constraint as the Ky-Fan norm to correct the online bias with SVD-free matrix completion. It used the majorization-minimization method to solve simple least squares.

The mobile coupon RS (Jooa et a., 2016) used distance and data analysis from GPS (Global Positioning System) to suggest local businesses of users' interest using a recommendation program and recommendation server. To suggest the book of buyer's interest, the book RS (Mathew et al., 2016) used equivalence class clustering & bottom-up Lattice Traversal (ECLAT) applied hybrid filtering and association rule mining. In this case, content-based RS filtered the entire book set based on the buyer's interest in purchasing and checking the purchased history from browsing data. It followed the steps of book dataset acquisition, pre-processing, transaction filtering, performing content and collaborative filtering methods, and final recommendation.

The automated graph-based music (Guo & Liu, 2016) generated and selected optimised meta-path-based features to rank the model and activate and eliminate the short sub-meta-paths at a low cost. Another trust-based RS (Jiang et al., 2018) illustrated the following steps: selecting trusted data, calculating the similarity between users, adding this similarity to the weight factor of the improved slope one algorithm, and lastly, getting the final recommendation. It found the system complexity for different parameters, such as difference (i, j) complexity was $O(n)$, similarity complexity was $O(m)$, and the complexity of slope one algorithm with trusted data was $O(mn^2)$. The complexity of slope one with the union of trusted data and similarity was $O(m^2n^2)$ for all items and users.

The review (Tripathi et al., 2016) on job recommender systems followed the steps of hashtags, applying entity resolution algorithm, canopy clustering the blocks and record linkage, and matching noisy records to clean the record. The book recommender (Kommineni et al., 2020) was created to assess the effectiveness of similarity measures in recommending books to users. Its steps were Merge Book Tags And Tags Dataset; Scrapped Some Tags To Obtain Genre Tags; Merged All Genre Tags And Authors For All Books; Distribute The Book Genre Tags Matrix Into Training And Testing Datasets; Extracted Strings Of Genres, And Authors; Applied TF-IDF vector to obtain tag matrix; applied the cosine similarity, Pearson correlation coefficient, constrained Pearson correlation, and JACCARD similarity measures on TAG matrix; and finally got top N suggestions based on a similarity matrix. The online education-based RS (Nikiforos et al., 2020) was designed to promote online courses and web-based learning material. The CF generated higher-quality suggestions for web-based learning platform users. The genetic algorithm determined the parameters that significantly impact the recommendation to improve the overall recommendation quality. Table 7 compares the problems focused on existing algorithms and their filtering type, data source, domain, data size, data type, and other related features and parameters. Table 8 discriminates the performance metrics, learning techniques, learning type, challenges, limitations, and future enhancements.

Table 7.

Comparing Problem Analysis, Filtering Type, Data Source and Domain, Data Size and Data Type, and Other Features of Existing Approaches - Survey 4

Reference	Problem Analysed	Filtering Type	Data Source & Domain	Data Size & Data Type	Other Features & Parameters
(Ojokoh et al., 2012)	Personalised Recommender Model	Content & Collaborative	CNET. Laptops.	Acer, Dell, HP, Sony & Toshiba. 50 Samples	Measured the similarity between user needs & product features.
(Kowalczyk et al., 2011)	To see the diverse effects of items, users & ratings on RS	-	Netflix. MovieLens.	Movie Ratings.	Used simulator for analysis.
(Christakopoulou & Karypis, 2014)	Recommender reducing prohibitive complexity through HOSLIM.	Content	Online websites. Real Item Datasets.	1200 item set with average size = 4. Items.	More higher-order relations lead toward recommendation quality improvement—less time.

(Gogna & Majumdar, 2015)	To validate online bias with RS.	Collaborative		Sufficient data set.	Used SFB-SVD method.
(Joa et al., 2016)	Recommender using Association Rules & CF.	Collaborative	Online websites. Mobile.	Ten users. Groceries & synthetic datasets.	Correlation coefficient = 0.712. Used the concept of personalisation.
(Mathew et al., 2016)	Design of efficient Book RS.	Content & Collaborative	Kaggle. Books.	Sufficient no. of Book Samples. 3 Members - Admin, Members/Registered User & Guest.	Used ECLAT to find frequent itemsets.
(Guo & Liu, 2016)	Heterogeneous graph-based Music Recommendation	Content	mxiami.com. Books.	1000 Samples & 50 Songs. 56055 Artists, 43086 Albums, 1233651 Songs, 633 Genres, 677275 Users, And 305916 Playlists.	472 features. Used dynamic parts have the supervised random walk algorithm to maximise the ranking performance
(Jiang et al., 2018)	Automated Trusted RS.	Collaborative	Amazon	Big-sized data.	K=3 & Trusted Ratio > 0.8. Combined trusted data & user similarity.
(Tripathi et al., 2016)	Job RS using entity resolution.	Collaborative	Skill Set Database. Job.	Skills like Java, Oops, C++, Visual Basic, C++, COBOL, C# & OO languages.	Job search at the right time with minimal effort and no missed opportunity—used multiple data modalities.
(Kommineni et al., 2020)	Book RS using ML to improve & fasten the process of purchasing items.	User-based Collaborative	Online Book Shopping. Good Reads book data.	A sufficient set of books.	Used similarity measure. Set angle-similarity parameters inversely proportional to each other.
(Nikiforos et al., 2020)	Online Education RS for higher quality suggestions for web-based learning platforms	Item-based & user-based CF	University website. Institution-based dataset.	A sufficient set of web learning data.	Very flexible system and optimum approach.

Table 8.

Discriminating Performance Metrics, Learning Techniques, Learning Type, Challenges, Limitations and Needs of Existing Approaches - Survey 4

Reference	Performance Metrics	Classification / Clustering / Prediction and Type Name	Challenges, Limitations and Needs
(Ojokoh et al., 2012)	Accuracy = 93%.	Classification, Fuzzy Logic.	Do not have the potential to handle increasing sales for online businesses.
(Christakopoulou & Karypis, 2014)	Very high accuracy. Better than the current best results.	SLIM, KNN, HOSLIM & HOKNN.	Extension with other ML techniques.
(Joa et al., 2016)	Promising Results.	Prediction. Association rules.	Need to enhance recommenders by providing relevant lists to both customers and businesses.
(Mathew et al., 2016)	Promising Results.	Clustering & Filtering.	Challenges: In the implementation, website development to sell the books, & RS module implementation based on user's interest. Coordination & implementation of hybrid content & collaborative filtering method. Trust towards the users—the system is used only for educated or knowledgeable people.

(Guo & Liu , 2016)	Promising Results.	Dynamic Feature Generation Tree.	Assumption: Each user has the same meta-path profile as the other—high computational cost.
(Jiang et al., 2018)	Precision is 31.9% more than other methods.	Prediction. Slope one algorithm using user similarity as a criterion.	Improve people's subjective behaviour of clicking on votes and identifying fraud internet users. Cold-start problem. Non-availability of user preference information.
(Tripathi et al., 2016)	Good Results.	Map-reduce clustering & matching.	Need to verify the information.
(Kommineni et al., 2020)	Recall, precision, F-score & Mean Absolute Precision of PCC, CPCC, Cosine & Jaccard.	User-User Similarity.	The data in the system must be protected from attacks, and other ways must be developed.
(Nikiforos et al., 2020)	Precision = 0.7 & recall = 0.2.	Prediction. Weight Discovering Genetic algorithms.	Need to evaluate the performance of the real system with actual users.

Therefore, it was found that most of the systems used collaborative and hybrid filtering methods with movie datasets. Most of them worked on clustering and predictive techniques for finding and suggesting recommendations to the users. These contributions evaluated many performance metrics, especially MAE and RMSE. They obtained good results, which need further improvements. The second factor is that many used limited or small-sized datasets. Thirdly, many of them are slow recommenders. So, there is a high need for an efficient and automated hybrid RS that can complete the recommendation process in significantly less time with a high level of performance.

Challenges and Issues

The last section illustrated four surveys and their gaps and limitations. These algorithms face many problems that drastically reduce their performance and efficiency. These problems are given as follows:

- Overwhelmed users are making poor decisions,
- Considering so many options to choose from in the datasets,
- Inadequate handling of data with imprecise information, as well as the gradualness of user preference,
- Cold start,
- Data sparsity,
- Grey sheep and over-specialisation,
- Finding Neighbors,
- Performance Metrics,
- Weighted, i.e., the weight of niche movies, is not factored into the prediction evaluation, and
- Feature augmentation.

Prominent One: Cold Start Problem

While launching a new product line, the expected behavior of a recommendation system is to recommend user items and other appropriately matched products, which provide increased visibility and reliability, which is the crux of the issue. With a simple recommendation system, the cold-start issue precludes the promotion of a genuine recommendation. According to the collaborative filtering approach, recommended engine rates are higher for new products, regardless of the user's preferences and interactions. Products with a greater degree of visibility sell better than rarely recommended products. This results in the popular recommendation system pushing things that sometimes do not suit the user. The same procedure applies to new users who have not yet purchased. The scenarios outlined above are referred to as product and user cold starts. When new items and consumers are introduced, a cold start happens. This main reason makes finding products and users a point of reference difficult.

Solution: If we start with product data, we can allocate products to specific categories, such as collections and descriptions, or product-specific qualities, such as size, color, or model. Considering all the features should give us a sense of the new product and its relationship to the existing, well-selling products. While acquiring data for items is very simple, gathering data to improve user features becomes more challenging. First, this is data that users must enter into the system when utilizing the platform. Examples of data include the user's location, login frequency, previously logged-in status, and usual transaction amounts. It becomes slightly more challenging when we attempt to collect behavioral data such as products viewed, device type, time spent on product detail pages, and average session duration.

The different types of data we have and the more choices we have while constructing the proposed solutions lead to better recommendation results. One brilliant example is the Light FM model, which Maciej Kula presents and combines any additional feature of numeric type with matrix factorization methods in a very intuitive and straightforward way.

Analytical Results

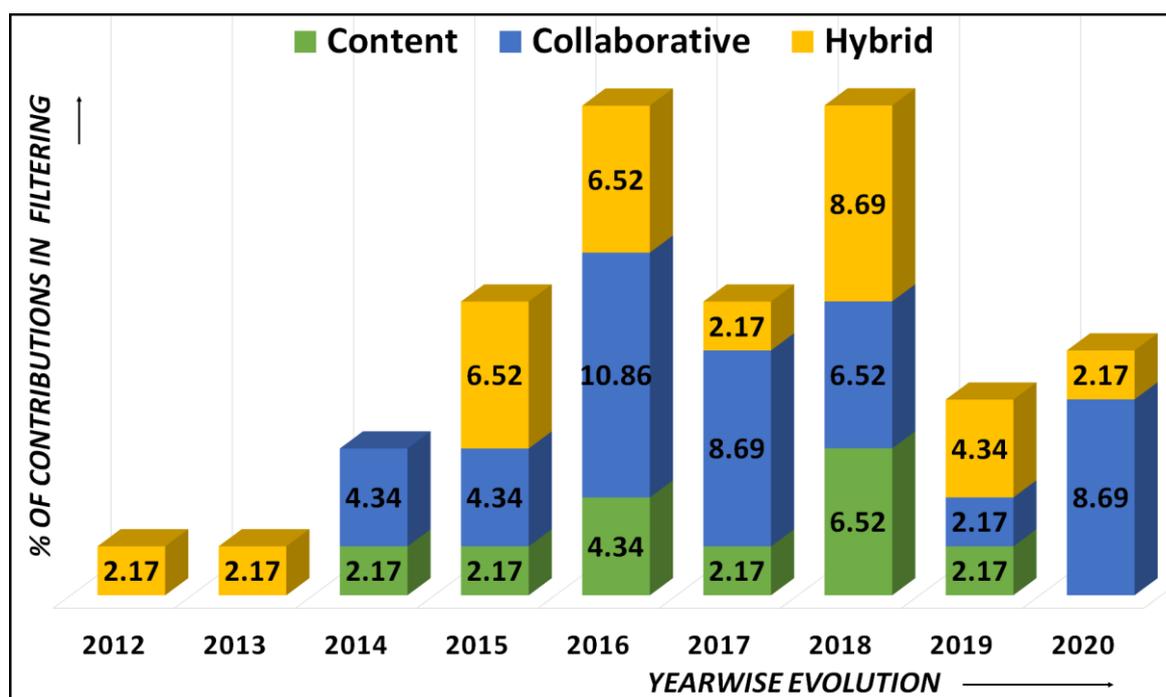
This section demonstrates various analyses of the existing systems, which depict their comparison graphically. The four sections include the research evolution of filtering techniques from 2012 to 2020, usage of learning paradigm, use of learning technique, and usage of online products, respectively.

Analysis 1: Year-Wise Analysis of Filtering Techniques

Analysis 1 shows the comparison among existing algorithms based on filtering method preference. These filtering methods are content, collaborative, and hybrid methods. It is shown in Figure 2 in the year-wise sequence.

Figure 2.

Showing the Research Evolution of Filtering Techniques from the year 2012 to 2020



Most research contributors have been observed to use hybrid and collaborative filtering as their preferred choice. The contributors preferred all three filtering methods in 2015, 2016, 2017, 2018, and 2019; the highest ratio of 8.69% was found for the hybrid method in 2018. Table 9 depicts the usage of different filtering paradigms for RS. Therefore, the deep learning-based hybrid method has a lot of scopes.

Table 9.

Depicting the Usage of Different Filtering Paradigms for RS

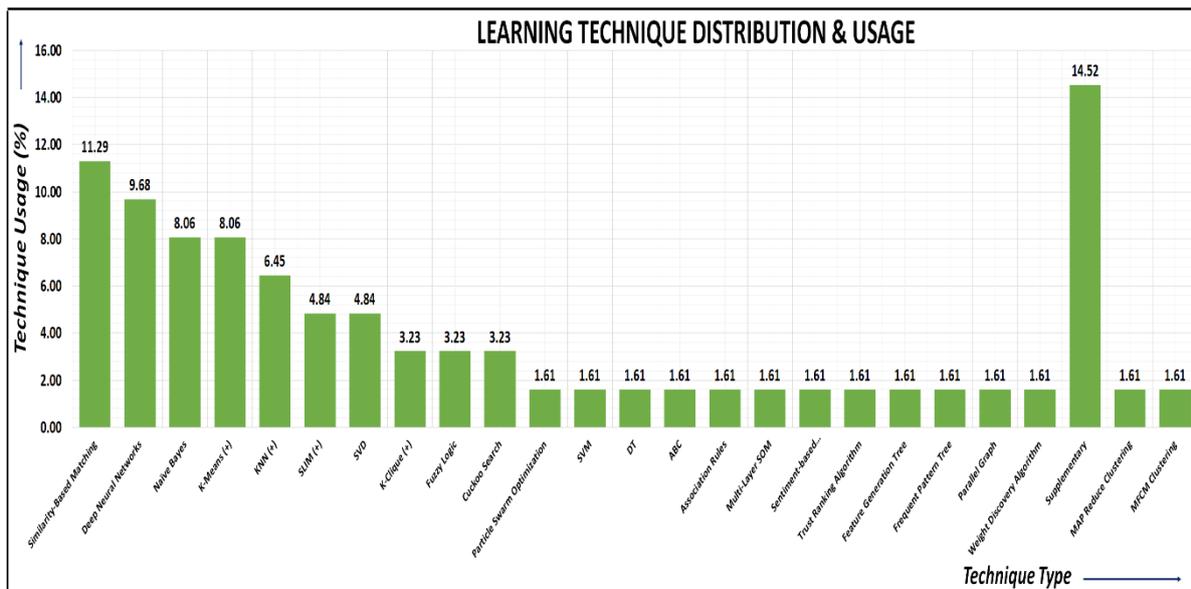
Year	Content Filtering	Collaborative Filtering	Hybrid Filtering
2012	×	×	✓
2013	✓	✓	×
2014	✓	✓	✓
2015	✓	✓	✓
2016	✓	✓	✓

2017	✓	✓	✓
2018	✓	✓	✓
2019	✓	✓	✓
2020	✗	✓	✓

Analysis 2: Usage of Learning Technique

Analysis 2 compares existing algorithms based on the learning technique used. It is shown in Figure 3 for various current approaches. They are given as Similarity-Based Matching, Deep Neural Networks, Naïve Bayes, K-Means (+), KNN (+), SLIM (+), SVD, K-Clique (+), Fuzzy Logic, Cuckoo Search, Particle Swarm Optimization, SVM, DT, ABC, Association Rules, Multilayer SOM, Sentiment-based Recommendation, Trust Ranking Algorithm, Feature Generation Tree, Frequent Pattern Tree, Parallel Graph, Weight Discovery Algorithm, Supplementary, MAP Reduce Clustering, and MFCM Clustering. Here '+' represents their version, enhancements, or improvements. It was observed that most contributors preferred similarity-based matching, with 11.29% usage. These similarity techniques include user-based, item-based, user-user-based, and fuzzy-based similarities. Other preferred techniques are Deep Neural Networks with 9.68%, Naïve Bayes with 8.06%, and K-Means (+) with 8.06%.

Figure 3.
Percentage Usage of Learning Techniques in Existing Approaches

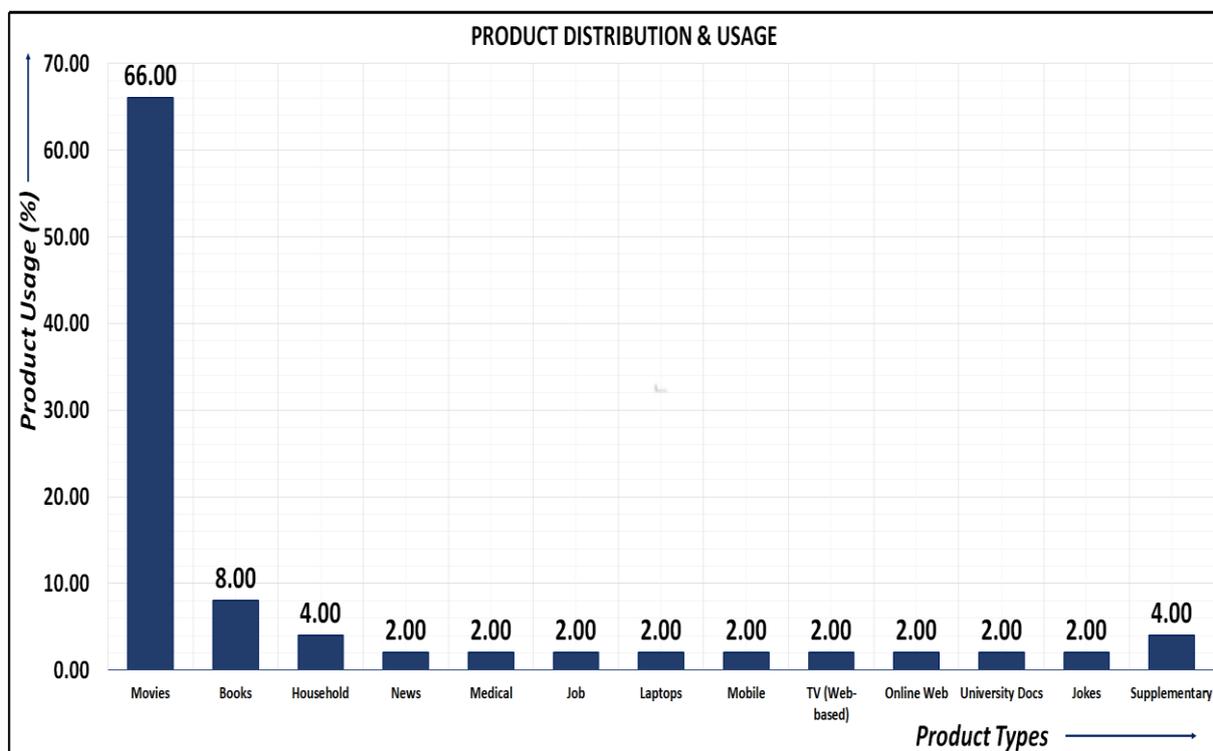


Analysis 3: Usage of Online Products

Analysis 3 depicts the existing algorithms' usage of different products, services, and items, for example, Movies, Books, Household items, News, Medical, Jobs, Laptops, Mobile, TV (Web-based), Online Web, University Docs, Jokes, and supplementary products. Most contributors worked on Movie datasets with 66% usage, books with 8% usage, and household items with 4% usage.

Figure 4.

Percentage Usage of Products in Existing Algorithms



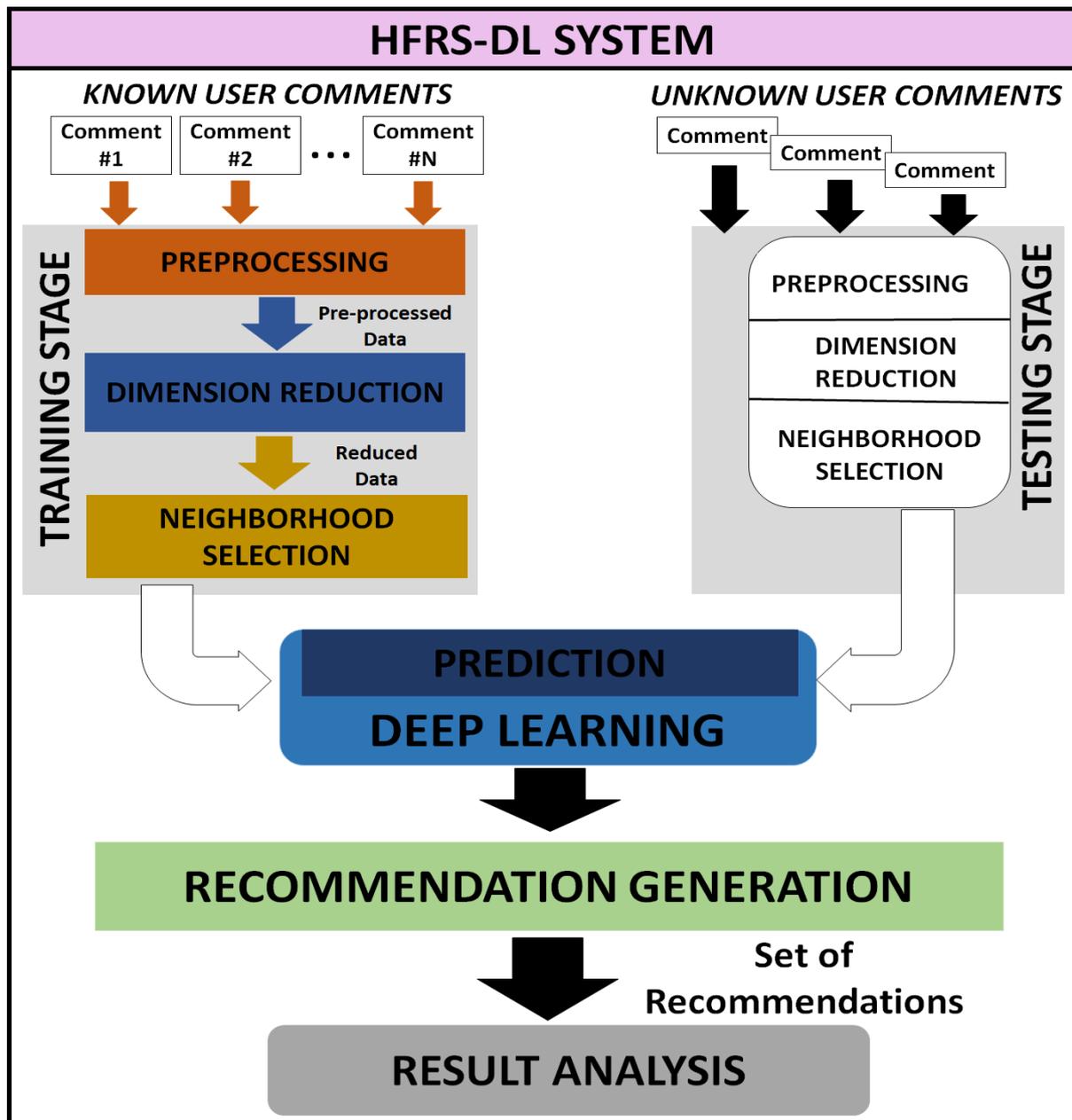
Methodology

The proposed hybrid filtering-based Recommendation System uses Deep Learning (HFRS-DL) system to recommend the suggestions of the movies to the user. Fig. 5 depicts the basic level design of the HFRS-DL system. This system design consists of two stages: the training and testing stages. A known set of comments collected from the IMDB sources are taken as inputs in the training stage. These N comments are Comment #1, Comment #2, ..., and Comment #N collected to acquire the data. After this, they are pre-processed and converted into a structured form. The sentences are extracted from the reviews and segmented into sentences using sentence tokenization.

Further, these dimensions are reduced using SVD. It is required to extract the features and perform the clustering. After selecting the neighborhood, these features are used to train the predictor. Lastly, the movie recommendations are generated and sent to the user.

Figure 5.

Proposed HFRS-DL Model



During the testing stage, unknown comments are collected from IMDB domains. They are first pre-processed and segmented, and then their features are extracted. Their dimensions are reduced, neighborhoods are selected, movies are recommended to the users, and results are analyzed.

Conclusion

This paper illustrated a detailed review of the existing algorithms of the deep learning-based hybrid recommenders and analyzers. This systematic survey was divided into four categories, and then their year-wise evolution and development were shown and compared. These contributions were differentiated based on problem analysis, filtering type, data source, data domain, data size, features, parameters, performance metrics, learning techniques, learning type, challenges, limitations and needs of existing approaches. Further, their challenges and issues elaborated gaps in accuracy, performance, cold and start problem, user preferences and satisfaction, and many more. These limitations need to be addressed and give birth to a new and efficient hybrid filtering-based recommendation system using deep learning. The proposed research effort can be expanded upon and investigated in various ways. This system can be implemented using reviews of multiple data domains such as service-based applications, social media, discussion forums, and so on. We can enhance this system by using other hybrid and multiple supervised learning approaches for its extension and implementation. Further, the system can be extended to handle emoticons, photos, and pictures included in comments.

Conflict of interest

The authors declare no potential conflict of interest regarding the publication of this work. In addition, the ethical issues including plagiarism, informed consent, misconduct, data fabrication and, or falsification, double publication and, or submission, and redundancy have been completely witnessed by the authors.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

References

- Aggarwal, C. C. (2016). Ensemble-based and hybrid recommender systems. In *Recommender systems* (pp. 199-224). Springer, Cham.
- Chen, L., Chen, G., & Wang, F. (2015). Recommender systems based on user reviews: the state of the art. *User Modeling and User-Adapted Interaction*, 25(2), 99-154.
- Christakopoulou, E., & Karypis, G. (2014). Hoslim: Higher-order sparse linear method for top-n recommender systems. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining* (pp. 38-49). Springer, Cham.
- Christakopoulou, E., & Karypis, G. (2016). Local item-item models for top-n recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems* (pp. 67-74).
- Gogna, A., & Majumdar, A. (2015). SVD free matrix completion with online bias correction for Recommender systems. In *2015 Eighth International Conference on Advances in Pattern Recognition (ICAPR)* (pp. 1-5). IEEE.

- Goyani, M., & Chaurasiya, N. (2020). A Review of Movie Recommendation System. *ELCVIA: electronic letters on computer vision and image analysis*, 19(3), 18-37.
- Gunjal, S. N., Yadav, S. K., & Kshirsagar, D. B. (2020). A hybrid scalable collaborative filtering-based recommendation system using ontology and incremental SVD algorithm. In *2020 International Conference on Smart Innovations in Design, Environment, Management, Planning and Computing (ICSIDEMPC)* (pp. 39-45). IEEE.
- Guo, C., & Liu, X. (2016). Dynamic feature generation and selection on heterogeneous graph for music recommendation. In *2016 IEEE International Conference on Big Data (Big Data)* (pp. 656-665). IEEE.
- Gurcan, F., & Birturk, A. A. (2016). A hybrid movie recommender using dynamic fuzzy clustering. In *Information Sciences and Systems 2015* (pp. 159-169). Springer, Cham.
- Harper, F. M., & Konstan, J. A. (2015). The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4), 1-19.
- Huanyu, M., Zhen, L., Fang, W., & Jiadong, X. (2016). Towards Efficient Collaborative Filtering Using Parallel Graph Model and Improved Similarity Measure. In *2016 IEEE 18th International Conference on High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS)* (pp. 182-189). IEEE.
- Indira, K., & Kavithadevi, M. K. (2019). Efficient machine learning model for movie recommender systems using multi-cloud environment. *Mobile Networks and Applications*, 24(6), 1872-1882.
- Jain, A., & Gupta, C. (2018). Fuzzy logic in recommender systems. In *Fuzzy Logic Augmentation of Neural and Optimisation Algorithms: Theoretical Aspects and Real Applications* (pp. 255-273). Springer, Cham.
- Jiang, L., Cheng, Y., Yang, L., Li, J., Yan, H., & Wang, X. (2019). A trust-based collaborative filtering algorithm for E-commerce recommendation system. *Journal of ambient intelligence and humanised computing*, 10(8), 3023-3034.
- Joa, J., Bangb, S., & Parka, G. (2016). Implementation of a recommendation system using association rules and collaborative filtering. *Procedia Computer Science*, 91, 944-952.
- Journal of High-Performance Computing and Networking*, 10(1-2), 54-63.
- Katarya, R. (2018). Movie recommender system with metaheuristic artificial bee. *Neural Computing and Applications*, 30(6), 1983-1990.
- Katarya, R., & Verma, O. P. (2016). A collaborative recommender system enhanced with particle swarm optimisation technique. *Multimedia Tools and Applications*, 75(15), 9225-9239.
- Katarya, R., & Verma, O. P. (2017). An effective collaborative movie recommender system with cuckoo search. *Egyptian Informatics Journal*, 18(2), 105-112.
- Kommineni, M., Alekhya, P., Vyshnavi, T. M., Aparna, V., Swetha, K., & Mounika, V. (2020). Machine learning based efficient recommendation system for book selection using user based collaborative filtering algorithm. In *2020 Fourth International Conference on Inventive Systems and Control (ICISC)* (pp. 66-71). IEEE.
- Kowalczyk, W., Szlavik, Z., & Schut, M. C. (2011). The impact of recommender systems on item-, user-, and rating-diversity. In *International Workshop on Agents and Data Mining Interaction* (pp. 261-287). Springer, Berlin, Heidelberg.
- Lin, W. W. (2017) Applying Artificial Neural Network to Deep Learning and Prescriptive Analysis in Telemedicine Systems using Microsoft Azure Machine Learning, *Amity Journal of Computational Sciences (AJCS)*, Vol. 1, issue 1, pp. 31-37.

- Liu, D., Wang, X., & Lu, H. (2014). Layered recommendation: A new strategy for movie promotion. In *2014 7th International Congress on Image and Signal Processing* (pp. 73-77). IEEE.
- Lu, L., & Zhang, H. (2015). A tree-structured representation for book author and its recommendation using multilayer SOM. In *2015 International Joint Conference on Neural Networks (IJCNN)* (pp. 1-8). IEEE.
- Mathew, P., Kuriakose, B., & Hegde, V. (2016). Book Recommendation System through content based and collaborative filtering method. In *2016 International conference on data mining and advanced computing (SAPIENCE)* (pp. 47-52). IEEE.
- Nagarnaik, P., & Thomas, A. (2015). Survey on recommendation system methods. In *2015 2nd International Conference on Electronics and Communication Systems (ICECS)* (pp. 1603-1608). IEEE.
- Nikiforos, M. N., Malakopoulou, M., Styliidou, A., Alvanou, A. G., Karyotis, V., & Kourouthanassis, P. (2020). Enhancing Collaborative Filtering Recommendations for Web-based Learning Platforms with Genetic Algorithms. In *2020 15th International Workshop on Semantic and Social Media Adaptation and Personalization (SMA)* (pp. 1-6). IEEE.
- Ojokoh, B. A., Omisore, M. O., Samuel, O. W., & Ogunniyi, T. O. (2012). A fuzzy logic based personalised recommender system. *International Journal of Computer Science and Information Technology & Security (IJCSITS)*, 2(5), 1008-1015.
- Peleja, F., Dias, P., Martins, F., & Magalhães, J. (2013). A recommender system for the TV on the web: integrating unrated reviews and movie ratings. *Multimedia systems*, 19(6), 543-558.
- Portugal, I., Alencar, P., & Cowan, D. (2018). The use of machine learning algorithms in recommender systems: A systematic review. *Expert Systems with Applications*, 97, 205-227.
- Sadanand, H., Vrushali, D., Rohan, N., Avadhut, M., Rushikesh, V., & Harshada, R. (2018). Movie recommender engine using collaborative filtering. In *Smart Computing and Informatics* (pp. 599-608). Springer, Singapore.
- Saipraba, N., & Subramaniaswamy, V. (2016). Enhancing stability of recommender system: an ensemble-based information retrieval approach. *Indian J. Sci. Technology*, 9, 48.
- Salam Patrous, Z., & Najafi, S. (2016). Evaluating prediction accuracy for collaborative filtering algorithms in recommender systems, pp. 1-37, 2016.
- Sattar, A., Ghazanfar, M. A., & Iqbal, M. (2017). Building accurate and practical recommender system algorithms using machine learning classifier and collaborative filtering. *Arabian Journal for Science and Engineering*, 42(8), 3229-3247.
- Selvi, C., & Sivasankar, E. (2019). A novel optimisation algorithm for recommender system using modified fuzzy c-means clustering approach. *Soft Computing*, 23(6), 1901-1916.
- Shokeen, J., & Rana, C. (2020). Social recommender systems: techniques, domains, metrics, datasets and future scope. *Journal of Intelligent Information Systems*, 54(3), 633-667.
- Subramaniaswamy, V., & Logesh, R. (2017). Adaptive KNN based recommender system through mining of user preferences. *Wireless Personal Communications*, 97(2), 2229-2247.
- Subramaniaswamy, V., Logesh, R., Chandrashekhar, M., Challa, A., & Vijayakumar, V. (2017). A personalised movie recommendation system based on collaborative filtering. *International Journal of High Performance Computing and Networking*, 10(1-2), 54-63.
- Tripathi, P., Agarwal, R., & Vashishtha, T. (2016). Review of job recommender system using big data analytics. In *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)* (pp. 3773-3777). IEEE.

- Tuan, S. Q., Sang, N. T. T., & Chau, D. T. H. (2018). An Effective FP-Tree-Based Movie Recommender System. In *Information Systems Design and Intelligent Applications* (pp. 172-182). Springer, Singapore.
- Vilakone, P., Park, D. S., Xinchang, K., & Hao, F. (2018). An efficient movie recommendation algorithm based on improved k-clique. *Human-centric Computing and Information Sciences*, 8(1), 1-15.
- Wang, H., Wang, N., & Yeung, D. Y. (2015). Collaborative deep learning for recommender systems. In *Proceedings of the 21st ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1235-1244).
- Wang, H., Zhang, F., Xie, X., & Guo, M. (2018). DKN: Deep knowledge-aware network for news recommendation. In *Proceedings of the 2018 world wide web conference* (pp. 1835-1844).
- Wang, Y., Wang, M., & Xu, W. (2018). A sentiment-enhanced hybrid recommender system for movie recommendation: a big data analytics framework. *Wireless Communications and Mobile Computing, 2018*.
- Wang, Z., Yu, X., Feng, N., & Wang, Z. (2014). An improved collaborative movie recommendation system using computational intelligence. *Journal of Visual Languages & Computing*, 25(6), 667-675.
- Wasid, M., & Ali, R. (2017). Use of soft computing techniques for recommender systems: an overview. *Applications of soft computing for the web*, 61-80.
- Zhang, L., Luo, T., Zhang, F., & Wu, Y. (2018). A recommendation model based on deep neural network. *IEEE Access*, 6, 9454-9463.
- Zhang, S., Yao, L., Sun, A., & Tay, Y. (2019). Deep learning-based recommender system: A survey and new perspectives. *ACM Computing Surveys (CSUR)*, 52(1), 1-38.

Bibliographic information of this paper for citing:

Ruchika; Sharma, Mayank & Hossain, Syed Akhter (2023). Efficient Machine Learning Algorithms In Hybrid Filtering Based Recommendation System. *Journal of Information Technology Management*, 15 (3), 134-161. [https://doi.org/ 10.22059/jitm.2023.93631](https://doi.org/10.22059/jitm.2023.93631)
