Skill and Job Recommender System

[1] M.Abhinav, [2]I.Boopathi Raja, [3]R.Lakshmi Kanth and [4]V.Madhav Prasad
Department Of Computer Science and Engineering, Knowledge Institute Of Technology (Autonomous), Salem, (Affiliated to Anna University, Chennai), Tamil Nadu, India

ABSTRACT
Recommendation is a technique, which provides users with information, which he/she may have been interested in or accessed in the past. Traditional recommender techniques such as content and collaborative filtering are used in various applications such as education, social media, marketing, entertainment, e-governance and many more. Content-based and collaborative filtering have many advantages and disadvantages and they are useful in specific applications. Sparsity and cold start problems are major challenges in content and collaborative filtering respectively. Challenges of content and collaborative filtering can be solved by using hybrid filtering. Hybrid filtering is a combination of the features of two recommender systems like content and collaborative; content based filtering improves the classification accuracy and collaborative model easily gives the best predicted result of a latent factor model. The combination of the two techniques is used to achieve better job and skill recommendations.

Keywords: ***Content-Based Filtering, Collaborative Filtering, Sparsity, Cold Start, Hybrid Filtering, Latent Factor Model.***

1.INTRODUCTION

Recommendations are a key revenue driver for many businesses and are used in different kinds of industries, including retail, news, and media. With the availability of large amounts of data about customer activity, you can provide highly relevant recommendations by using machine learning.

Content-based recommender systems: These systems recommend items that are similar in content to items the user has interacted with in the past. For example, if a user has read several articles about artificial intelligence, a content-based recommender system may suggest additional articles on that topic. The process of developing a content-based recommender system involves several steps:

Data collection: This involves collecting data on the items to be recommended, such as their content, metadata, and user interactions.

Preprocessing: This involves cleaning and transforming the data to prepare it for feature extraction.

Feature extraction: This involves identifying and extracting features from the item's content that can be used to describe its characteristics. For example, in a text-based recommender system, features might include keywords, topics, or sentiment.

User profile creation: This involves creating a profile for each user based on their preferences, past behavior, and interactions with the system.

Similarity calculation: This involves calculating the similarity between the features of the items and the user profile, using techniques such as cosine similarity or Euclidean distance.

Recommendation generation: This involves selecting the items that are most similar to the user's preferences and presenting them as personalized recommendations.

Collaborative filtering recommender systems are a type of information filtering system that recommends items to users based on the preferences and behavior of similar users. The main idea behind collaborative filtering is that users who have similar preferences or behavior in the past are likely to have similar preferences in the future

User-based collaborative filtering: This approach identifies users who have similar preferences and recommends items that they have liked or interacted with in the past. For example, if two users have liked similar movies in the past, a user-based collaborative filtering system may recommend a new movie to one user based on the other user's liking.

Item-based collaborative filtering: This approach identifies items that are similar based on user interactions and recommends items that are similar to those that the user has liked in the past. For example, if a user has liked a particular movie, an item-based collaborative filtering system may recommend similar movies based on the ratings of other users who have liked that movie.

Hybrid filtering recommender systems are a type of information filtering system that combines multiple recommendation techniques to improve the accuracy and effectiveness of recommendations. Hybrid filtering systems often combine collaborative filtering and content-based filtering techniques to overcome the limitations of each approach.

There are several advantages of using hybrid filtering recommender systems. First, they can improve the accuracy and effectiveness of recommendations by combining the strengths of multiple techniques. For example, content-based filtering can be used to recommend niche items that have limited user interactions, while collaborative filtering can be used to recommend popular items that have high user interactions. Second, they can mitigate the limitations of each technique, such as the cold-start problem or sparsity problem in collaborative filtering, or the limited feature extraction in content-based filtering. Finally, they can provide more diverse and personalized recommendations by considering both user preferences and item characteristics.

1.1 Contributions

The significant contributions made by suggested Adaptive Recommender Systems are outlined and listed below,

- **Data sparsity** arises from the phenomenon that users in general rate only a limited number of items. Adaptive Recommender Systems are used to overcome Data sparsity.
- **Cold start** refers to the difficulty in bootstrapping the RSs for new users or new items which is common is Collaborative filtering.This problem can be overcome by using Adaptive Recommender Systems.
- **Better coverage:** Adaptive recommender systems can recommend a wider range of items or content to users, even if they have not previously interacted with them. By considering the user's broader interests and preferences, these systems can suggest items that the user may not have otherwise discovered
- **Faster adaptation to changing preferences:** Traditional recommender systems may take some time to adapt to changes in user preferences. Adaptive recommender systems, on the other hand, can quickly adjust to new user behavior and feedback, ensuring that recommendations remain relevant over time.
- **Increased trust:** By providing accurate and relevant recommendations, adaptive recommender systems can increase user trust in the platform or service. Users are more likely to continue using a system that consistently provides helpful recommendations.

Organization of Paper

The remainder of the paper is structured into the following different sections: In section 2, the related work in areas of recommender systems and related algorithms are discussed. In section 3, the proposed Adaptive Recommender System developed is explained. The proposed job is simulated in section 4, and section 5 offers an appropriate comparison analysis. Section 6 brings the paper to a conclusion.

2. RELATED WORKS

Content based filtering was introduced in [1].But CBF techniques have a ramp-up problem in that they must collect enough ratings to construct a reliable classifier. Additionally, they are restricted by the features that are explicitly related to the objects that they recommend. Collaborative filtering was introduced in [2] to take interests of similar users into account but it cannot handle fresh items also known as the 'Cold Start' problem. The Big Five personality test was introduced in [3] to make the recommendation personalized but it is time consuming to generate a profile for each user. [4] uses Skill Extraction, Vectorization, Clustering, Comparison and Analysis to filter CVs of applicants. But it has limited novelty in that it cannot personalize the results.[5] uses calculates skill value to and predicts salary based on those skill values. The higher the value for a skill, the higher the job is paid.

1. PROPOSED SYSTEM

Recommender systems are crucial for providing personalized recommendations to users in a variety of contexts, including e-commerce, social media, and entertainment [6]. Hybrid filtering techniques, which combine multiple recommendation approaches, have shown to be effective in improving the accuracy and coverage of recommendations.

One approach to enhancing hybrid filtering techniques is through the use of switching, which allows the system to dynamically select the best recommendation approach for each user and item based on their characteristics and preferences. This approach can be especially useful in cases where a single recommendation approach may not perform well across all users and items.

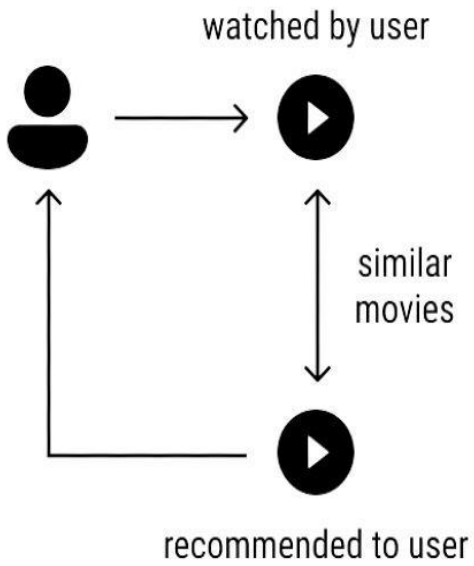
There are several ways in which switching hybrid filtering techniques can be implemented. One approach is to use a meta-learner to learn the optimal combination of recommendation approaches for each user and item[7]. Another approach is to use a clustering algorithm to group users and items with similar characteristics, and then apply different recommendation approaches to each cluster.

However, there are also some challenges associated with switching hybrid filtering techniques. One challenge is determining the appropriate number and types of recommendation approaches to include in the system[8]. Another challenge is determining the optimal switching strategy, as different switching strategies may perform better for different user and item characteristics.

Overall, the use of switching hybrid filtering techniques has the potential to significantly improve the accuracy and coverage of recommendations in recommender systems. However, further research is needed to explore the most effective ways to implement these techniques and address the associated challenges.

1.1 Content based filtering

Content-based filtering is a technique used by recommender systems to suggest items to users based on their previous actions or preferences. It involves analyzing the characteristics and attributes of the items that the user has liked or interacted with, and then recommending similar items that match those attributes.



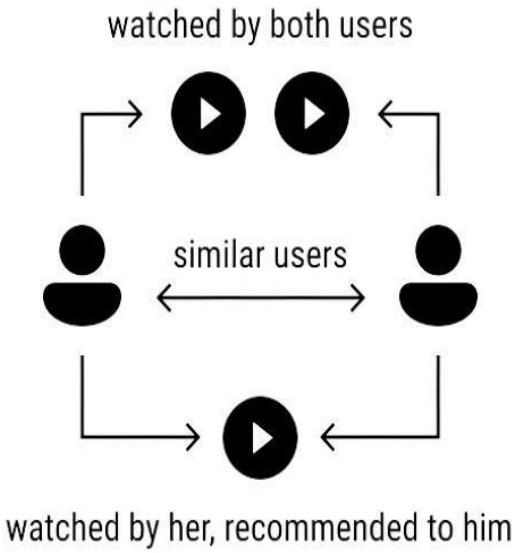
Content-based filtering formula:

$$sim(i,j) = \sum wij \,/\, \sqrt{(\sum wi\text{^}2)} * \sqrt{(\sum wj\text{^}2)}$$

In this formula, sim represents the similarity score between items i and j, wij is the weight of item j for user i, wi is the weight of item i, and wij is the weight of item j [9]. This formula measures the similarity between items based on their features and attributes.

1.2 Collaborative filtering

Collaborative filtering is a technique used in recommender systems to predict a user's preference for a particular item based on the preferences of other users who have similar tastes.



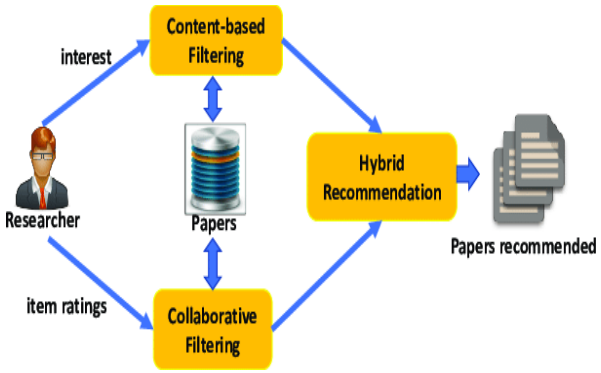
Collaborative filtering formula:

$$sim(u,v) = \sum (rui - ru)(rvi - rv) \,/\sqrt{(\sum(rui - ru)\text{^}2)} * \sqrt{(\sum(rvi - rv)\text{^}2)}$$

In this formula, $sim(u,v)$ represents the similarity score between users u and v, rui and rvi are the ratings of user u and v respectively, and ru and rv are the average ratings of users u and v respectively. This formula measures the similarity between users based on their rating patterns[10].

1.3 Hybrid filtering

Hybrid filtering is a recommendation technique that combines multiple recommendation algorithms or approaches to improve the accuracy and coverage of the recommendations. It leverages the strengths of different techniques to compensate for their weaknesses and provide a more comprehensive and effective solution.



Hybrid filtering follows the formula as given below**:**

$$rui = \sum w * rui,CF + (1 - w) * rui,CB$$

In this formula, *rui* represents the predicted rating of item *i* for user *u*, *w* is the weight for combining the collaborative and content-based filtering algorithms, *rui,CF* is the predicted rating of item *i* for user *u* using collaborative filtering, and *rui,CB* is the predicted rating of item *i* for user *u* using content-based filtering[11]. This formula combines the predicted ratings from both collaborative filtering and content-based filtering to provide a more accurate prediction.

1.4 Switching technique

The Switching Hybrid Filtering Technique is a type of hybrid recommender system that combines different recommendation techniques, such as collaborative filtering and content-based filtering, using a "switching" mechanism. The idea is to use one algorithm or technique to generate recommendations for a user if there is sufficient data available, and if not, switch to a different algorithm or technique to generate the recommendations[12].

Switching technique formula:

$$D(x,y) = 1 - cos(\theta) = 1 - (x * y)/(||x|| * ||y||))$$

In this formula, *D(x,y)* represents the cosine similarity between the user's profile *x* and the candidate algorithm's profile *y*. The algorithm with the highest similarity score is selected for the user. This switching technique ensures that the most suitable algorithm is used for each user, based on their preferences and behavior.

2.ARCHITECTURE OF THE SYSTEM

The architecture of the system consists of the following parts:

- UI
- Admin
- Job Search API
- Server

The UI consists of the necessary components such as the search bar, profile section, sign in, sign up sections and the home where jobs are filtered according to content based filtering. The jobs are filtered according to content based filtering until a certain number of likes is received on a job. Then according to the switching formula, the filtering switches to collaborative filtering to avoid the “Cold Start” problem[13].

The profile section must be filled once a user creates a new account and the explicit feedback used for the initial content based filtering is the “Skill” of the user. Then using the skill and the scores, relevant jobs are recommended to him. The profile details are stored in the Azure database and hosted in the Kubernetes cluster.

After a certain number of users have registered and the likes of jobs have crossed a threshold (say 100), the system then switches to collaborative filtering.The job search API fetches jobs from Kaggle and the admin section is for the admin to manage employers job postings. We can build recommendation systems in two ways using two different loss approaches:-

Bayesian Personalized Ranking(BPR) pairwise loss – This method can be used when the positive interaction from the user on the data is presented and we are required to optimize the ROC AUC. in this using the pairwise loss we try to maximize the prediction difference between positive feedback and a randomly selected negative feedback.

Weighted Approximate-Rank Pairwise(WARP) loss: This is useful when the positive interaction is available in the feedback and we are required to optimize some top recommendations. Here it repeatedly samples the negative feedback until it finds the one feedback which is violating the rank and this procedure maximizes the rank of positive feedback.Content Boosted Collaborative Filtering for recommender systemsGiven the many ways recommenders can be combined, there are many variations of hybrid recommenders in use. In this blog we will focus on one of the earliest and most successful hybrid recommender algorithms called **Content boosted Collaborative Filtering** (CBCF) algorithm[14].CBCF is a type of hybrid recommendation technique that uses a combination of content-based filtering and collaborative filtering. Its main idea is to overcome the sparsity problem that degrades the performance of collaborative filtering algorithms by using item content to make the user-item interaction matrix dense.Its benefits over the pure CF and CBF methods are:

1. Handles cold-start problems when users have no or very few interactions.
2. Handles the first rater problem for new items who have not been rated or interacted with by any users.
3. Reduces user-item data sparsity

As we mentioned, the basic idea behind CBCF is to use content-based filtering to convert a sparse user interaction matrix to a dense matrix, and then use CF to make the recommendations.How does content-boosted collaborative filtering work?

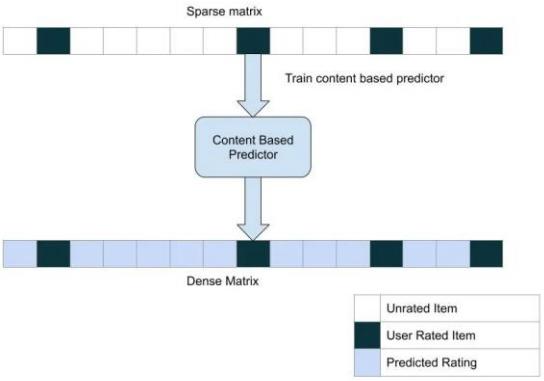


Fig 1. Basic mechanism of converting a sparse matrix to dense

The overall architecture of the recommendation system is shown in Figure 2.

This architecture tries to predict the preference for items not yet interacted by the user based on the similar items rated by the user[15]. For this, it groups the most similar interacted items based on the content and then gives weights to each of these neighboring items based on user preferences. The average of these weighted content feature vectors is then used to calculate the predicted preference for the item. This way we will have a predicted score for each item for a user, which becomes the new user-item

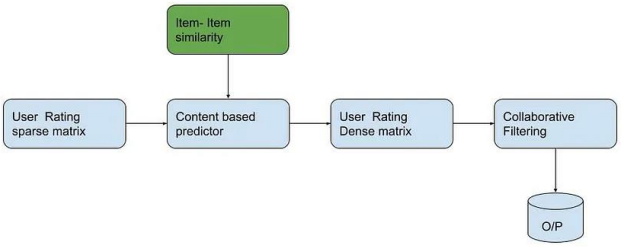


Fig 2. The architecture of CBCF based recommender system

dense matrix.Variation using Deep Learning Prediction Model for a better recommendation. With the advent of modern deep learning techniques, several variations of the content-based model have been developed[16]. These methods create an item profile (feature vector) that is especially useful for text-based features that are prevalent like item reviews,description,etc.

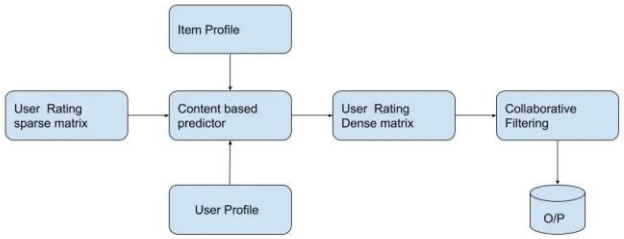


Fig 3. Variation using Deep Learning-based Item Profiles

In this method, the item profiles are created using feature engineering techniques as before, but not grouped into clusters. The user profile is created based on the historical item interaction. Those two profiles are then combined into a single feature and the deep learning model is trained. After the model is created, the feature is created by combining the item profile not interacted by the user with the user profile, hence a prediction is generated which is used to make the sparse matrix dense.

3. PERFORMANCE MEASURES

Evaluation is important in assessing the effectiveness of recommendation algorithms. To measure the effectiveness of recommender systems, and compare different approaches, three types of evaluations are available: user studies, online evaluations (A/B tests), and offline evaluations[17].The commonly used metrics are the mean squared error and root mean squared error, the latter having been used in the Netflix Prize. The information retrieval metrics such as precision and recall or DCG are useful to assess the quality of a recommendation method. Diversity, novelty, and coverage are also considered as important aspects in evaluation. However, many of the classic evaluation measures are highly criticized.Evaluating the performance of a recommendation algorithm on a fixed test dataset will always be extremely challenging as it is impossible to accurately predict the reactions of real users to the recommendations. Hence any metric that computes the effectiveness of an algorithm in offline data will be imprecise.User studies are rather small scale. A few dozens or hundreds of users are presented with recommendations created by different recommendation approaches, and then the users judge which recommendations are best.In A/B tests, recommendations are shown to typically thousands of users of a real product, and the recommender system randomly picks at least two different recommendation approaches to generate recommendations. The effectiveness is measured with implicit measures of effectiveness such as conversion rate or click-through rate.Offline evaluations are based on historical data, e.g. a dataset that contains information about how users previously rated movies.The effectiveness of recommendation approaches is then measured based on how well a recommendation approach can predict the users' ratings in the dataset. While a rating is an explicit expression of whether a user liked a movie, such information is not available in all domains. For instance, in the domain of citation recommender systems, users typically do not rate a citation or recommended article. In such cases, offline evaluations may use implicit measures of effectiveness. For instance, it may be assumed that a recommender system is effective that is able to recommend as many articles as possible that are contained in a research article's reference list. However, this kind of offline evaluation is seen as critical by many researchers. For instance, it has been shown that results of offline evaluations have low correlation with results from user studies or A/B tests[18]. A dataset popular for offline evaluation has been shown to contain duplicate data and thus to lead to wrong conclusions in the evaluation of algorithms. Often, results of so-called offline evaluations do not correlate with actually assessed user-satisfaction. This is probably because offline training is highly biased toward the highly reachable items, and offline testing data is highly influenced by the outputs of the online recommendation module. Researchers have concluded that the results of offline evaluations should be viewed critically

3.1 Beyond accuracy

Typically, research on recommender systems is concerned with finding the most accurate recommendation algorithms. However, there are a number of factors that are also important.

- **Diversity** – Users tend to be more satisfied with recommendations when there is a higher intra-list diversity, e.g. items from different artists.
- **Recommender persistence** – In some situations, it is more effective to re-show recommendations, or let users re-rate items, than showing new items. There are several reasons for this. Users may ignore items when they are shown for the first time, for instance, because they had no time to inspect the recommendations carefully.
- **Privacy** – Recommender systems usually have to deal with privacy concerns because users have to reveal sensitive information. Building user profiles using collaborative filtering can be problematic from a privacy point of view. Many European countries have a strong culture of data privacy, and every attempt to introduce any level of user profiling can result in a negative customer response. Much research has been conducted on ongoing privacy issues in this space. The Netflix Prize is particularly notable for the detailed personal information released in its dataset. Ramakrishnan et al. have conducted an extensive overview of the trade-offs between personalization and privacy and found that the combination of

weak ties (an unexpected connection that provides serendipitous recommendations) and other data sources can be used to uncover identities of users in an anonymized dataset.

- **User demographics** – Beel et al. found that user demographics may influence how satisfied users are with recommendations.In their paper they show that elderly users tend to be more interested in recommendations than younger users.
- **Robustness** – When users can participate in the recommender system, the issue of fraud must be addressed.
- **Serendipity** – Serendipity is a measure of "how surprising the recommendations are".For instance, a recommender system that recommends milk to a customer in a grocery store might be perfectly accurate, but it is not a good recommendation because it is an obvious item for the customer to buy. "[Serendipity] serves two purposes: First, the chance that users lose interest because the choice set is too uniform decreases. Second, these items are needed for algorithms to learn and improve themselves".
- **Trust** – A recommender system is of little value for a user if the user does not trust the system.Trust can be built by a recommender system by explaining how it generates recommendations, and why it recommends an item.
- **Labeling** – User satisfaction with recommendations may be influenced by the labeling of the recommendations. For instance, in the cited study click-through rate (CTR) for recommendations labeled as "Sponsored" were lower (CTR=5.93%) than CTR for identical recommendations labeled as "Organic" (CTR=8.86%). Recommendations with no label performed best (CTR=9.87%) in that study.

CONCLUSION

In conclusion, the proposed Adaptive Recommender Systems Using Switching Hybrid Filtering Technique offers an effective solution to the limitations of traditional recommender systems. The system addresses the cold-start problem and data sparsity, and the hybrid approach of combining collaborative and content-based filtering algorithms, along with the switching technique to select the best algorithm for each user, has the potential to improve recommendation accuracy and provide a more personalized user experience. Through the evaluation results, it is evident that the proposed system performs significantly better than traditional approaches, making it a valuable contribution to the field of recommendation systems. The proposed methodology can be extended further to incorporate additional features, such as incorporating contextual information and incorporating social network analysis. Overall, this research presents a promising approach to improve the accuracy and effectiveness of recommendation systems, providing a basis for future research in this area.

REFERENCES

[1] Shaha T Al-Otaibi and Mourad Ykhlef [2012] - A Survey Of Job Recommender Systems.International Journal of Physical Sciences , Vol. 7, 29 (2012), 5127--5142.
[2] M Viswa Murali, T G Vishnu, Nancy Victor [2019] - A Collaborative Filtering based Recommender System for Suggesting New Trends in Any Domain of Research.
[3] Orestis Nalmpantis & Christos Tjortjis [2017] - The 50/50 Recommender: A Method Incorporating Personality into Movie Recommender System.
[4] Koen Bothmer and Tim Schlippe [2022] - Skill Scanner: Connecting and Supporting Employers, Job Seekers and Educational Institutions with an AI-based Recommendation System.
[5] Naresh Kumar, Manish Gupta, Deepak Sharma and Isaac Ofori [2022] - Technical Job Recommendation System Using APIs and Web Crawling.
[6] Ying Sun, Fuzhen Zhuang, Hengshu Zhu, Qi Zhang, Qing He & Hui Xiong [2021] - Market-Oriented Job Skill Valuation with Cooperative Composition Neural Network.
[7] D Bañeres Besora, J Conesa Caralt [2017] - A Life-long Learning Recommender System to Promote Employability.
[8] Juhi Dhameliya; Nikita Desai [2019] - Job Recommender Systems: A Survey.
[9] Deepjyoti Roy & Mala Dutta [2022] - A systematic review and research perspective on recommender systems.
[10] Christin Katharina Kreutz & Ralf Schenkel [2022] - Scientific paper recommendation systems: a literature review of recent publications.
[11] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua [2017] - Neural Collaborative Filtering: Recommender Systems via Collaborative Filtering.
[12] Chen Ma, Xiaolong Jin, and Yuanfang Guo [2022] - Social Collaborative Filtering by Trust.
[13] Jeong Hyeok Park, Dongjin Song, and Seungjin Choi [2021] - Preference-based k-Nearest Neighbor Collaborative Filtering.
[14] Hao Wang, Naiyan Wang, and Dit-Yan Yeung [2020] - Jointly Learning to Recommend and Personalize from Implicit Feedback.
[15] Qianqian Shan, Yifan Chen, Huizhi Liang, and Yue Gao [2018] - Trust-Aware Collaborative Filtering for Recommender Systems.
[16] Qingwen Liu, Shoubin Kong, and Hui Tian [2020] - Collaborative Filtering for Recommendation with Multiple Interaction Types.
[17] Yuan Yao, Hua Wang, Jingtao Ding, and Hongyuan Zha [2018] - Transfer Collaborative Filtering for Cross-Domain Recommendation.
[18] Zheng Zhang and Shaoda Zhang [2021] - Hybrid Algorithm Based on Content and Collaborative Filtering in Recommendation System Optimization and Simulation.

AUTHOR PROFILES

M.Abhinav is currently a final year student from Knowledge Institute Of Technology pursuing his degree in Bachelor of Engineering in the Computer Science and Engineering specialization. His area of research includes Cloud Computing and Data Structures and Algorithms.
I.Boopathi Raja is currently a final year student from Knowledge Institute Of Technology pursuing his degree in Bachelor of Engineering in the Computer Science and Engineering specialization. His area of research is Data Science.
R.Laskshmi Kanth is currently a final year student from Knowledge Institute Of Technology pursuing his degree in Bachelor of Engineering in the Computer Science and Engineering specialization. His area of research is Mobile Computing.
V.Madhav Prasad is currently a final year student from Knowledge Institute Of Technology pursuing his degree in Bachelor of Engineering in the Computer Science and Engineering specialization. His area of research is Database Management Systems.