IJARCCE
ISSN (O) 2278-1021, ISSN (P) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁘ Peer-reviewed & Refereed journal ⁘ Vol. 15, Issue 6, June 2026
DOI: 10.17148/IJARCCE.2026.156104

AniRec : Personalized Anime Recommendation System Using Hybrid Filtering

Sudarshan Mane[1], Mr. Sagar Mahajan[2], Vedant Kadole[3], Mohit Tiwari[4]

Undergraduate Student, Dept. of Information Technology, G.H. Raisoni College of Engineering and Management, Pune, India[1]

Assistant Professor, Dept. of Information Technology, G.H. Raisoni College of Engineering and Management, Pune, India[2]

Undergraduate Student, Dept. of Information Technology, G.H. Raisoni College of Engineering and Management, Pune, India[3]

Undergraduate Student, Dept. of Information Technology, G.H. Raisoni College of Engineering and Management, Pune, India[4]

Abstract: The popularity of anime continues to rise rapidly, thus making it difficult to locate what one is interested in. With thousands of animes in a variety of genres, people tend to spend too much time looking for something worth watching, leading them to end up picking something trendy instead. AniRec, on the other hand, takes into consideration your preferences without you realizing it. Using innovative techniques, it analyses your preferences and suggests anime titles suitable for you. AniRec works on how people watch while also breaking down what shows are like. Starting off, the platform gathers details about each anime and gets them ready for deeper review. Gaps in info get filled, odd entries fixed so everything lines up right. Once tidy, things like story kind, score, or format get pulled out and turned into codes that allow the system to compare different anime effectively. By using features of each show, comparisons begin. Instead of relying on user ratings, the method checks how much one series looks like another in themes, genres, or art style. By measuring angles between data points, Cosine Similarity finds which titles sit closest in trait space. Once scored, rankings form; higher values mean stronger links to what the viewer liked before. What appears at the front of results reflects tight alignment in characteristics. Seen first in the browser view, those picks guide eyes toward fresh options shaped by personal taste. Python is used for system's backend, with Flask. Pandas handles data work while calculations rely on NumPy. Machine learning steps come alive through Scikit-learn tools. Stored inside the system's database are things like what users rate, choose, watch. Because of these details, suggestions get better bit by bit, learning exactly what each person likes. Overall, AniRec provides you personalized anime recommendation as per your preference and previous records.

Index Terms: Anime Recommendation, Cosine Similarity, Machine Learning, Personalization, Hybrid Filtering.

I. INTRODUCTION

Streaming services are growing bigger and bigger, changing how everyone perspective how to watch shows and anime; anime is attracting more and more people. Because of variety in styles and stories, picking something feels endless. Though having more to watch sounds good, it can leave viewers stuck, unsure what fits their taste. Too much choice leads some to scroll for ages, others settle for suggestions they don't like after watching it halfway. [5], [10]

One reason many anime sites feel repetitive because they rely on what's popular or currently trending. Instead of digging into what you actually like, they just show top charts. Because of this, two people who enjoy completely different stories end up choosing same anime to watch. This repetition makes the navigation boring. An intelligent recommendation engine, such as AniRec, should use the data on viewing behavior. This system would make recommendations not generic and rather personal. [4], [11]

AniRec recommends anime based on users behavior and interests. Due to learning by users' viewing behavior and preferences, the engine becomes more efficient with each additional use. As the users interact with the engine, the results adapt to their preferences, and there is no need for further input. Gradually, it eliminates any unnecessary recommendations, which means simpler navigation and less distraction during the viewing. The process takes place in the background and is not noticeable to the user; however, only specific recommendations based on the previous interaction are shown. [6], [12], [13].

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⵣ Peer-reviewed & Refereed journal ⵣ Vol. 15, Issue 6, June 2026
DOI: 10.17148/IJARCCE.2026.156104

AniRec builds custom anime suggestions using patterns found through machine learning. Because it checks details like genre, peoples review, episode count, and if it's popular, are used while recommending. Following a pick or score given by someone, the tool lines up matches shaped around what they've liked before. Rather than using past data from many users, it uses traits of the anime itself. This means new profiles get useful picks right away, skipping delays that occur when starting out. [2], [3]

One main aim behind AniRec is making it easier to find anime, which also boosts how involved users feel along with their overall experience [5], [11]. AniRec helps people pick shows; AniRec also provides a section that helps keep track of anime you are watching [22].

II. OBJECTIVE

Main objective of AniRec: Personalized Anime Recommendation System Using Hybrid Filtering is to provide personal recommendation based on your views and likes. Solve cold start problem by providing generic and trending anime. AniRec system collect data from website to improve itself, also provide personal space where you can manage watched, watching, plan to watch anime.

III. LITERATURE REVIEW

Recommendation systems have become one of the important component of modern digital platforms, helping users navigate through large amount of content by providing personalized suggestions. We have had a number of different types of recommendation algorithms over the years, such as collaborative filtering, content based filtering, and hybrid techniques which were studied and implemented [5], [11].

Early research in recommendation systems primarily focused on collaborative filtering (CF) techniques. User based and item based collaborative filtering methods analyze user behavior and preferences to recommend items. According to Resnick et al., CF systems rely on the belief that users that had similar interest scores in the past will presumably like those items again [23]. But these methods suffer issues such as cold start, data sparsity and scalability [24]. To overcome these limitations, content based filtering (CBF) approaches were introduced. These systems recommend items based on item attributes and user profiles. Pazzani and Billsus demonstrated that content-based methods are effective in domains where item metadata is available [3]. However, such systems often lack diversity and may lead to overspecialization, where users are only recommended items similar to their past preferences [14].

Hybrid recommendation systems combine both collaborative and content based techniques to improve performance. Burke categorized hybrid systems into multiple types, such as weighted, switching, and mixed hybrid models [4]. These systems aim to balance personalization and diversity while mitigating the limitations of individual methods. With the advancement of machine learning, matrix factorization techniques gained popularity.

Koren et al. introduced latent factor models that decompose user item interaction matrices to discover hidden patterns [2]. These models significantly improved recommendation accuracy and were widely adopted in platforms like Netflix. Recent research has shifted towards deep learning based recommendation systems. Neural networks, including autoencoders and recurrent neural networks (RNNs), have been used to capture complex user-item relationships. Additionally, attention mechanisms and embedding techniques allow systems to better understand user preferences and item similarities [6], [9], [13].

For anime recommendation systems, there have been researches done on certain domain related problems, including genre, episodes, and evolving user interests. Most of the platforms are dependent on popularly recommended anime, but this method fails to recognize the individual taste of a user [5], [16]. It is essential to consider factors like behavior, rating, and filtering by genre to provide more accurate recommendations. Additionally, another critical issue is the cold start problem, which occurs when the recommender system does not have sufficient information about a new user and the newly introduced anime [19], [24]. To overcome the problem, researchers have suggested using demographic recommendation and hybrid recommender systems [16]. In this process, demographics of the user like age, gender, and past behavior help recommend anime.

The recommended system designed in this project follows the same technique and introduces user preference analysis, demographic filtering, and interaction-based learning. As compared to traditional methods, the system uses recommendation logic rather than popularity measures to provide recommendations to the users [1], [17].

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ∺ Peer-reviewed & Refereed journal ∺ Vol. 15, Issue 6, June 2026

DOI: 10.17148/IJARCCE.2026.156104

IV. METHODOLOGY

AniRec is an intelligent recommendation engine specifically designed to provide personalized recommendations for anime. Rather than making any subjective assumptions or rules, the system works intelligently to make decisions using the patterns identified in actual anime data [6], [12]. The approach employed by AniRec provides a detailed roadmap for building an intelligent recommendation engine for anime using machine learning. The design of each step is meant to leverage the previous step, thereby ensuring accuracy, scalability, stability, and adaptability [5], [17].

A. Data Collection and Preprocessing

Obtained anime information from credible and accessible sources. The dataset consists of anime name, genre, type, number of episodes, rating, and popularity. These multiple features aid in defining various aspects of anime. Enhanced the quality of data and reliability of the model; missing values for genre are replaced with blank and missing values for rating are replaced with average rating [1], [5].

B. Feature Extraction

All anime are represented as a binary vector:

$$X_i = [x_1, x_2, \ldots, x_n] \quad (1)$$

where $x_k = 1$ if the anime belongs to genre k, and $x_k = 0$ otherwise [3].

C. Similarity Calculation and Calculation of Popularity Score

Similarities between anime are calculated based on their cosine similarity, which evaluates the distance between two genre vectors [2], [3]:

$$Sim(i, j) = (A \cdot B) / (|A| \times |B|) \quad (2)$$

where A and B represent the binary vectors of genres of anime i and j, and $A \cdot B$ refers to dot products while $|A|$ and $|B|$ to Euclidean distances.

This enables recognition of anime with similar genre compositions. The higher the similarity scores, the more similarities there are between anime titles. Similarity score values range from 0 to 1; a score of 1 indicates identical genre composition and 0 no similarities [2].

To consider quality and trends in user preferences, a popularity score is calculated, combining rating and member numbers through normalized values. First, normalization is performed by Min-Max Normalization:

$$R'_i = (R_i - R_{min}) / (R_{max} - R_{min}) \quad (3)$$

where R_i represents the rating of anime i, M_i represents the number of members, while min and max indicate the minimum and maximum values among all the animes in the dataset. This transformation makes both variables to lie within the range of [0, 1]. Finally, the popularity score is computed based on the following formula:

$$S_i = 0.6 \cdot R'_i + 0.4 \cdot M'_i \quad (4)$$

where the weights of 0.6 and 0.4 are used to determine the importance of normalized rating and member count. Thus, the final score not only considers the popularity of each anime but also takes into account its rating [18].

To integrate the notion of quality and trend of user preferences, a popularity score for each anime is calculated by applying Min-Max normalization to its rating and member count [5].

D. Recommendation Generation

The recommendation process follows these steps:

- User selects an anime
- System retrieves its corresponding feature (genre) vector
- Cosine similarity is computed between the selected anime and all other anime [2], [3]
- Anime with higher similarity scores are identified as similar candidates
- Popularity score for each candidate anime is considered (based on rating and member count) [18]
- Final score is calculated by combining similarity and popularity: $FinalScore(j) = \alpha \cdot Sim(i, j) + (1 - \alpha) \cdot S_j$ [4]
- All anime are ranked in descending order as per the final score [25]

ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 6, June 2026

DOI: 10.17148/IJARCCE.2026.156104

• Top-N most relevant anime are selected [25]
• Recommended anime are displayed to the user

V. SYSTEM ARCHITECTURE

Fig. 1. System architecture of the AniRec recommendation system. The user interacts with the React.js frontend, while Appwrite manages authentication and user preferences. The recommendation engine processes user data and anime features from the dataset using hybrid filtering to generate personalized recommendations [4], [6]. The TMDB API is used to fetch anime metadata, and the final results are presented to the user with feedback used for future improvement [22].

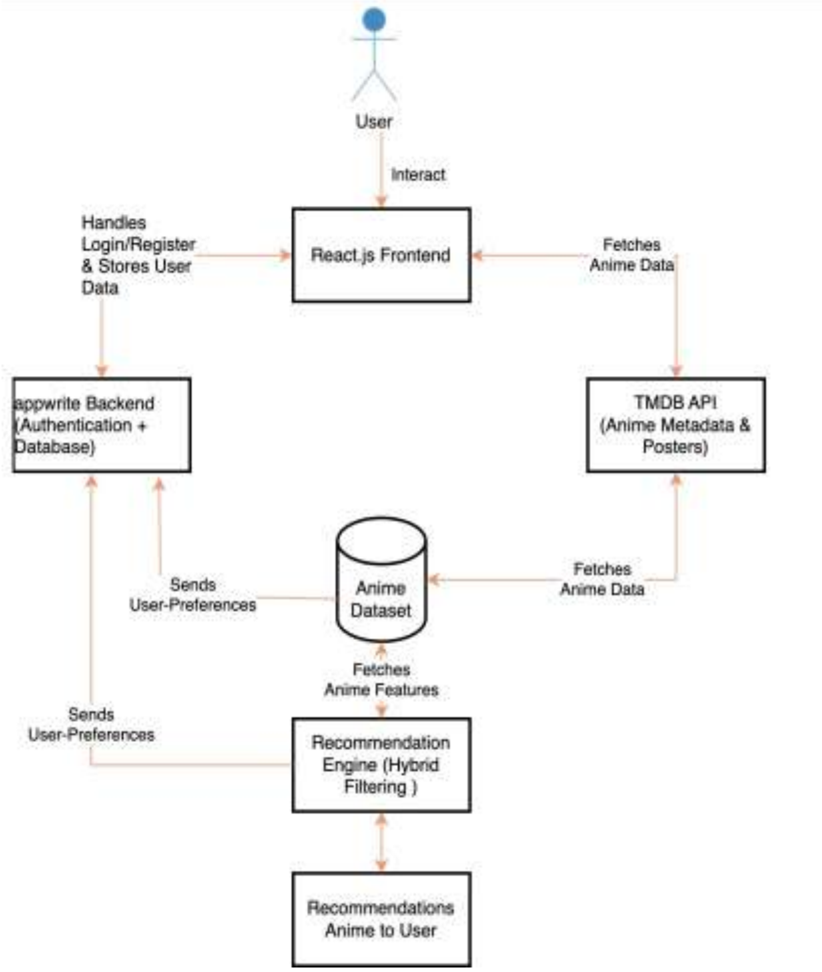

Fig. 1. Architecture Diagram

VI. IMPLEMENTATION DETAILS

AniRec: Personalized Anime Recommendation System Using Hybrid Filtering was implemented using a dataset containing anime titles, genres, ratings, and user engagement (members). The system integrates Hybrid based filtering with popularity based ranking to generate personalized recommendations.

Home Page: The system follows a modular architecture, combining a React.Js/Tailwind CSS/Vite frontend for user interface as shown in fig. 2.

Authentication: For Authentication Appwrite is used. It eliminates need to build backend Authentication from scratch. Appwrite provide feature like User Login, Session Management, Secure user data handling.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 6, June 2026

DOI: 10.17148/IJARCCE.2026.156104

VII. RESULTS AND DISCUSSION

AniRec Personalized Anime Recommendation System was implemented using a dataset containing anime titles, genres, ratings, and user engagement on website. The system integrates content-based filtering with popularity-based ranking with views and like of users to generate personalized recommendations for user.

A. Precision@K Analysis

The Precision@K test was performed to check how efficient the recommendation is depending on the value of K. The following conclusion can be drawn based on the obtained results: the most accurate recommendations are characterized by low K values. In other words, the first recommendations offered to the user match his/her preferences the most. As the value of K increase it leads to the decline in the precision because irrelevant items are added to the recommendations.

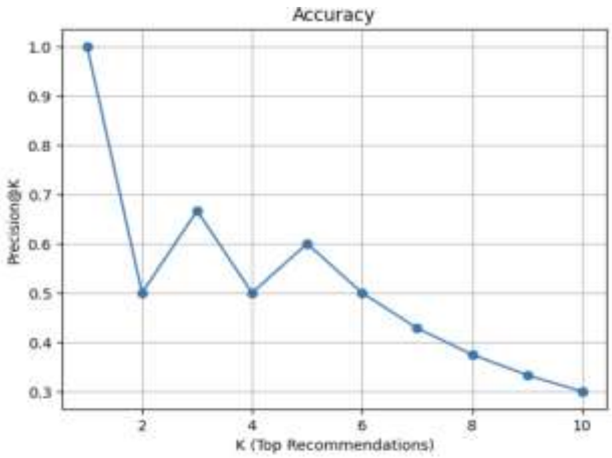

Fig. 2. Accuracy Graph

B. Demographic Filtering

Results of demographic filtering refers to how your system gives recommendation based on basic user preferences; it's also known as traditional system. It provides solution for cold start problem by giving basic recommendation.

C. Performance of Hybrid Approach

Content based filtering ensures personalization while popularity based ranking ensures quality; combining both systems provides better results. As shown in fig. 4, the Hybrid system provides better results compared to traditional system. Recall is described as how many related items you found while F1 score combines both metrics: recall and precision.

Popularity score is computed using rating; it improves recommendation quality because of this low quality and unpopular anime are filtered out.

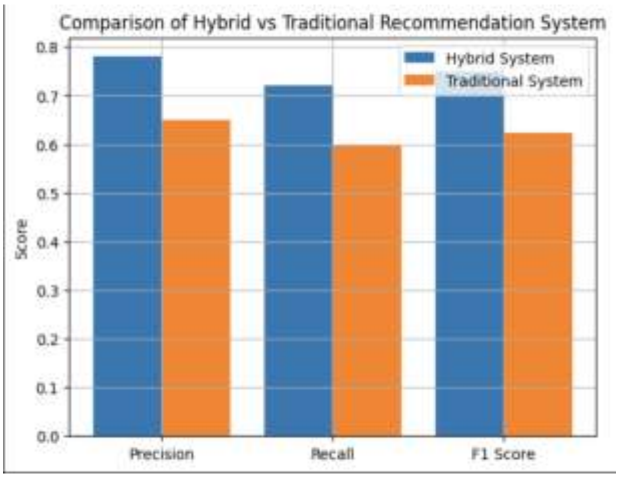

Fig. 3. Hybrid vs Traditional Recommendation System

ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⧖ Peer-reviewed & Refereed journal ⧖ Vol. 15, Issue 6, June 2026

DOI: 10.17148/IJARCCE.2026.156104

D. Experimental Results

The content-based method generates recommendations on similar anime using their genre analysis. For this purpose, the system utilizes the cosine similarity matrix, enabling the detection of similar anime according to their feature vectors. As a result, the recommendations will reflect the genre and content of the selected anime.

In addition to user-based personalized recommendations, the website will have some predefined categories, including Top Airing, Most Popular, Most Favourite, and Latest Completed. These categories will contribute positively to improving user experience and solving the cold start problem by generating valuable recommendations even in the absence of user-related data. Furthermore, these categories will facilitate users search for recognized and popular anime among other anime.

Fig. 4. Home Page

Moreover, the Spotlight category with anime recommendations will assist users in finding trending anime. The users will be able to identify popular anime rapidly without conducting additional research. Additionally, the website will include the Trending Anime category, which will represent the most viewed anime currently. This approach will provide users with access to the most popular anime.

VIII. FUTURE SCOPE

There are numerous possibilities for improving the AniRec system concerning accuracy, scalability, and user experience. The current hybrid recommendation algorithm can be further improved by applying advance machine learning algorithms like deep learning and neural collaborative filtering in order to describe more complex relations between anime and users [6], [13].

Real time recommendations can be generated by automatically adjusting recommendations depending on user activities and responses [20]. Also, additional data such as ratings, watchlists, and other social parameters can be further enhanced and be more accurate for recommendation generation [9].

Scalability can be increased by applying cloud computing that can help you with data handling and data security [1]. Additionally, not only The Movie Database (TMDB) API, but also some other external APIs, can be integrated into the platform to deliver more accurate metadata. Mobile applications can be developed in order to enhance system popularity, make it easy operate on multiple platform [20].

IX. CONCLUSION

The AniRec: Personalized Anime Recommendation System Using Hybrid Filtering. This project successfully demonstrates design and implementation of a personalized anime recommendation system [4]. With the rapid growth of anime content across digital platforms, users often face difficulty in discovering relevant titles that match their interests [5], [10]. This problem is solved by AniRec through provision of data driven, precise recommendations that make it easy to discover content. The AniRec system utilizes a well defined and modular framework that starts from data collection and processing and then moves on to extraction of features and the construction of content based recommendations models [3]. Through analysis of attributes of animes like their genre and ratings, among others, AniRec is able to find similarities between different animes and provide relevant recommendations. This helps AniRec to solve cold-start problems [19], [24].

ISSN (O) 2278-1021, ISSN (P) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁒ Peer-reviewed & Refereed journal ⁒ Vol. 15, Issue 6, June 2026
DOI: 10.17148/IJARCCE.2026.156104

Throughout the development process, emphasis was placed on data quality, scalability, and system flexibility [1], [17]. The modular structure allows changes in the individual modules without any disruption to the system as a whole. Thus, AniRec remains amenable to modifications such as hybrid recommendation systems and others [6], [12].

ACKNOWLEDGMENT

The authors would like to thank the faculty and fellow students of G H Raisoni College of Engineering and Management for their valuable guidance and feedback.

REFERENCES

[1] F. Ricci, L. Rokach, and B. Shapira, Recommender Systems Handbook, 2nd ed. New York, NY, USA: Springer, 2015.
[2] Y. Koren, R. Bell, and C. Volinsky, “Matrix factorization techniques for recommender systems,” Computer, vol. 42, no. 8, pp. 30–37, Aug. 2009.
[3] M. J. Pazzani and D. Billsus, “Content-based recommendation systems,” in The Adaptive Web: Methods and Strategies of Web Personalization, Berlin, Germany: Springer, 2007, pp. 325–341.
[4] R. Burke, “Hybrid recommender systems: Survey and experiments,” User Modeling and User-Adapted Interaction, vol. 12, no. 4, pp. 331–370, 2002.
[5] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, “Recommender systems survey,” Knowledge-Based Systems, vol. 46, pp. 109–132, Jul. 2013.
[6] S. Zhang, L. Yao, A. Sun, and Y. Tay, “Deep learning based recommender system: A survey and new perspectives,” ACM Computing Surveys, vol. 52, no. 1, pp. 1–38, 2019.
[7] Z. Zhang et al., “Deep learning techniques for recommender systems,” IEEE Access, 2019.
[8] X. Amatriain, J. M. Pujol, and N. Oliver, “I like it... I like it not: Evaluating user ratings noise in recommender systems,” in Proc. 17th Int. Conf. User Modeling, Adaptation, and Personalization (UMAP), 2009, pp. 247–258.
[9] Z. Zhang et al., “Deep learning techniques for recommender systems,” IEEE Access, 2019.
[10] X. Amatriain, J. M. Pujol, and N. Oliver, “I like it... I like it not: Evaluating user ratings noise in recommender systems,” in Proc. 17th Int. Conf. User Modeling, Adaptation, and Personalization (UMAP), 2009, pp. 247–258.
[11] G. Adomavicius and A. Tuzhilin, “Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions,” IEEE Trans. Knowl. Data Eng., vol. 17, no. 6, pp. 734–749, Jun. 2005.
[12] H. Wang, N. Wang, and D.-Y. Yeung, “Collaborative deep learning for recommender systems,” in Proc. 21st ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2015, pp. 1235–1244.
[13] H. Wang et al., “Collaborative deep learning combining neural networks and CF,” IEEE Trans. Pattern Anal. Mach. Intell., 2019.
[14] P. Lops, M. de Gemmis, and G. Semeraro, “Content-based recommender systems: State of the art and trends,” in Recommender Systems Handbook, New York, NY, USA: Springer, 2011, pp. 73–105.
[15] J. Schafer, D. Frankowski, J. Herlocker, and S. Sen, “Collaborative filtering recommender systems,” in The Adaptive Web, Berlin, Germany: Springer, 2007, pp. 291–324.
[16] K. Haruna et al., “Context-aware recommender system: A review of recent developmental processes and future research direction,” Applied Sciences, vol. 8, no. 11, p. 2220, Nov. 2018.
[17] C. C. Aggarwal, Recommender Systems: The Textbook. Cham, Switzerland: Springer, 2016.
[18] J. Davidson et al., “The YouTube video recommendation system,” in Proc. 4th ACM Conf. Recommender Systems (RecSys), 2010, pp. 293–296.
[19] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, “BPR: Bayesian Personalized Ranking from Implicit Feedback,” in Proc. UAI, 2009.
[20] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, “LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation,” in Proc. SIGIR, 2020.
[21] S. Sun, Y. Zhang, W. Zhang, and Y. Yu, “A survey on context-aware recommender systems,” IEEE Access, vol. 8, pp. 110746–110763, 2020.
[22] A. Covington, J. Adams, and E. Sargin, “Deep neural networks for YouTube recommendations,” in Proc. RecSys, 2016.
[23] P. Resnick and H. R. Varian, “Recommender systems,” Communications of the ACM, vol. 40, no. 3, pp. 56–58, 1997.
[24] J. L. Herlocker, J. A. Konstan, and J. Riedl, “Evaluating collaborative filtering recommender systems,” ACM Trans. Inf. Syst., vol. 22, no. 1, pp. 5–53, 2004.
[25] M. Deshpande and G. Karypis, “Item-based top-n recommendation algorithms,” ACM Transactions on Information Systems, vol. 22, no. 1, pp. 143–177, 2004.