Grenze International Journal of Engineering and Technology, Jan Issue

Personalized Music Recommendation System: Combining Collaborative Filtering and Semantic Analysis

Deepali Naik[1], Varunraj Tipugade[2], Pranav Kulkarni[3], Aditi Shinde[4] and Ashootosh Pawar[5]
[1-5]Department of Computer Engineering, Pimpri Chinchwad College of Engineering, Pune, India
Email: deepali.naik@pccoepune.org, varunrajtipugade2001@gmail.com, pranav.kulkarni20@pccoepune.org, aditishinde035@gmail.com, ashootoshpawar14@gmail.com

***Abstract*— This research paper introduces a novel personalized music recommendation system that leverages collaborative filtering and semantic analysis techniques. The system aims to generate customized music recommendations tailored to individual users' preferences and listening histories. By employing collaborative filtering, the system identifies users with similar music tastes and suggests music they enjoy. Additionally, semantic analysis is utilized to analyze the current sentiment of a user, enabling the system to recommend songs suitable for the user's mood for initial recommendations. A dataset containing user listening histories and music metadata is utilized to evaluate the system's performance. The results demonstrate that the proposed system surpasses traditional collaborative filtering approaches in identifying the cold-start issue for new users. This capability enhances the music listening experience by providing more relevant and personalized recommendations while facilitating the discovery of new and captivating music. The system's unique approach of combining collaborative filtering and semantic analysis provides a fresh perspective on personalized music recommendations, contributing to the advancement of music recommendation systems. The evaluation outcomes signify the promising nature of the proposed method. Future research can focus on further optimizing the system's performance through additional refinements and evaluations.**

***Index Terms*— Collaborative filtering, User correlation, Recommendation, Rating Matrix, Singular value decomposition, root-mean-square-error, Data Sparsity, Cold Start.**

I. Introduction

The exponential growth of streaming services such as Spotify, Apple Music, and Amazon Music has provided consumers access to an extensive collection of songs worldwide. While this abundance of music has made it easier to discover new artists and genres, it has also presented a challenge in finding the right music that resonates with each individual. To address this challenge, music recommendation systems employ advanced algorithms that analyze user behavior and preferences, leveraging this information to offer tailored recommendations for new artists, albums, and songs.

Personalized music recommendations are crucial in enhancing the consumer experience and benefiting the music industry [1]. By leveraging sophisticated algorithms and data analysis, these systems enable users to discover music that aligns with their unique tastes, expanding their musical horizons beyond what they would have found independently [2]. Additionally, these recommendations help artists reach new audiences, facilitating the

Grenze ID: 01.GIJET.10.1.75

discovery and dissemination of music in a mutually beneficial manner.
Collaborative filtering is one of the most popular methods in music recommendation systems. This approach assumes that users with similar musical preferences will likely appreciate songs and artists that align with those preferences. Collaborative filtering generates recommendations based on these shared interests by analyzing user listening behavior and identifying patterns and similarities [3]. For example, if a user enjoys a particular music subgenre, the system can identify listeners with similar tastes and suggest new music they will likely enjoy. Collaborative filtering effectively addresses the challenge of delivering necessary recommendations across different user preferences [4]. However, collaborative filtering algorithms encounter a cold start problem when faced with new users [3]. The absence of information about these users hinders the system's ability to provide accurate recommendations. Traditional approaches rely on historical user data to generate recommendations, making it challenging to effectively address the cold start problem.
This research suggests a unique method incorporating semantic analysis to address the cold start issue in music recommendation systems. Semantic analysis, a field within artificial intelligence and natural language processing, focuses on comprehending the meaning and context of human language. The semantic analysis unveils underlying meaning by analyzing text, enabling its application in multiple domains involving music recommendation systems. We leverage semantic analysis [5] to analyze user mood as an additional factor for music recommendation. By examining the music by their semantic content in the lyrics that express specific moods or emotions, catering to users seeking music with particular qualities. This approach enables us to provide recommendations based on the user's mood or emotional preferences, even without historical listening behavior. By combining semantic analysis with collaborative filtering techniques, we aim to enhance the accuracy and relevance of recommendations, particularly for new users who lack a substantial listening history.
This research paper presents our methodology and findings on incorporating semantic analysis to solve the cold start problem in music recommendation systems. By combining collaborative filtering with semantic analysis of user mood, we aim to address the challenge of providing accurate and relevant music recommendations for new and existing users. Our approach holds the potential to enhance the user experience, foster music discovery, and contribute to the growth of the music industry in the era of abundant music streaming services.
The research paper follows a structured format comprising various sections. The introduction provides an overview of the topic, states the research problem, and establishes the study's significance. The literature review critically examines existing research, identifying key theories, concepts, and gaps in the literature. The methodology section describes the research design, data collection methods, and analytical techniques. The mathematical model section presents the developed model, including variables, equations, and explanations. The evaluation section assesses the research's validity and presents the obtained results. The discussion section interprets the findings, relates them to the research objectives, and explores their implications. The conclusion summarizes the main findings, restates the objectives, and mentions any limitations or future research areas. The bibliography lists all the cited sources, ensuring proper attribution.

II. Literature Review

Music recommendation systems have gained significant attention due to the availability of vast music libraries and the need to assist users in discovering music that aligns with their preferences. Several approaches and algorithms have been explored to address the challenges of personalized music recommendations, including collaborative filtering, content-based filtering, sentiment analysis, and the utilization of social tags.
Collaborative filtering is a widely used method in music recommendation systems. It leverages the behavior and preferences of users to identify patterns and similarities, recommending music based on other users' likes and dislikes with similar tastes. Various algorithms, such as UserKNN, UserKNNAvg, NMF, and CoClustering, have been applied to generate accurate recommendations [6]. This approach effectively addresses the cold start problem for existing users but requires historical user data for exact recommendations.
Content-based filtering is another approach employed in music recommendation systems. It focuses on the perceptual properties of music objects, extracting features and grouping music based on them. The system utilizes user preferences from access histories to provide personalized music recommendations [7]. Content-based filtering analyzes recent user interests to suggest music belonging to similar groups, enabling recommendations that align with a user's recent preferences [8]. This approach differs from collaborative filtering, which relies on the behavior of other users with similar preferences.
Sentiment analysis has also been integrated into music recommendation systems. By analyzing user comments, ratings, and critics using models like BERT, sentiment analysis helps identify user sentiment and preferences. This information is then used in conjunction with collaborative filtering to generate recommendations [1]. Incorporating

sentiment analysis enhances the relevance and accuracy of recommendations, as user sentiment provides valuable insights into their preferences.

The use of social tags offers an alternative perspective for improving music recommendations. Traditionally used in content-based techniques, social tags can be applied to describe user expertise rather than the music itself. Analyzing the tags users assign to items and applying TF-IDF analysis makes it possible to infer a user's level of specialization. The information is then used to improve algorithms for user-based collaborative filtering, improving the quality of recommendations [9]. Comparing the similarity between users based on the feedback of more expert users can lead to more relevant recommendations [10].

Overall, the existing literature highlights the diverse approaches employed in music recommendation systems, including collaborative filtering, content-based filtering, sentiment analysis, and the utilization of social tags. These approaches contribute to personalized music recommendations, addressing challenges such as the cold start problem and user preferences and incorporating user sentiment and expertise. By combining these approaches and algorithms, researchers strive to enhance the accuracy and relevance of music recommendations, ultimately improving the user experience and fostering music discovery.

III. THE METHODOLOGY

Our music recommendation system utilizes two main models: the BERT model and collaborative filtering. Figure 1 presents the methodology of the proposed music recommendation system. The BERT model processes the user's responses to a set of questions and performs natural language processing to analyze the user's mood. This initial analysis forms the basis for the initial song recommendations provided by the system. The BERT model employs transformers and pipelines to efficiently process the text and extract relevant information [11]. Following the initial recommendations, the collaborative filtering model takes over. It utilizes a co-occurrence matrix that captures item-based similarity to generate further song recommendations. The collaborative filtering model considers the History of the user consuming songs and identifies other users with similar preferences to suggest songs they may enjoy. The model utilizes a user rating matrix to make accurate recommendations, which stores the listening count of songs and the number of users who have listened to them.

The system architecture involves several steps, including data collection, processing, semantic analysis, and the recommendation engine. Users provide their responses through a questionnaire, and the collected data undergoes pre-processing using the BERT model. This pre-processing involves removing slang, grammatical errors, and stop words to enhance the quality of the input. Semantic analysis is then performed to determine the user's mood numerically.

The recommendation engine combines the BERT model's analysis results and the collaborative filtering model's predictions to generate personalized song recommendations. The proposed system's database stores comprehensive song information, including listening and user counts, which the collaborative filtering model utilizes for accurate recommendations.

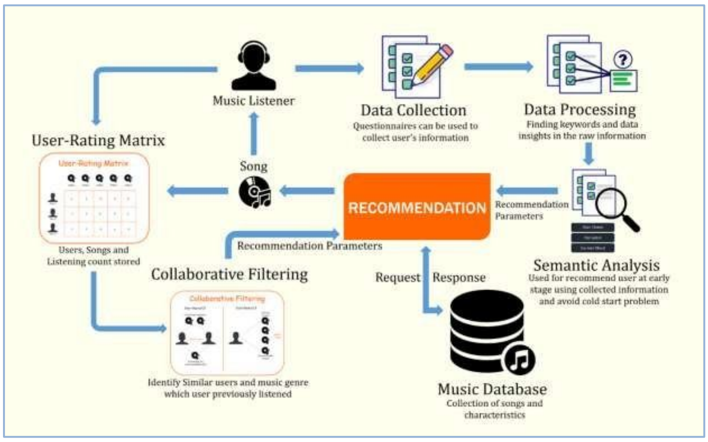

Figure 1: Methodology of Proposed Music Recommendation System

IV. MATHEMATICAL MODEL

The collaborative filtering algorithm applying a recommender system consists of popularity-based and item similarity-based recommendations. The class for music recommendations is based on how well-liked the songs are. A song's popularity is determined by the number of times it has been listened to by users. The more a song is listened to, the higher its popularity. In this class, the top 20 popular songs are recommended to users based on their overall popularity.

The item similarity-based recommendation class utilizes collaborative filtering techniques. It starts by initializing the user id, item id, and co-occurrence matrix. The co-occurrence matrix captures item similarity based on user activity and preferences.

To recover similar items, the collaborative filtering model solves the co-occurrence matrix problem. It begins by obtaining a list of all unique songs and all unique users. The model identifies songs with similar user interaction patterns by analyzing the co-occurrence matrix. These matching items are then recommended to users interested in a particular song.

Overall, the collaborative filtering model combines popularity-based and item similarity-based recommendations to provide users with diverse and personalized song recommendations.

Let,

Co-occurrence matrix represented as "C" _"[Si][Sj]"

"S" represents THE POOL OF SONGS.

"U" represents THE POOL OF SONGS.

Hence,

|S| represents unique songs, |U| represents unique users

Return all the users associated with all unique songs |S|.

Initialize the co-occurrence matric of size len (|S|) x len (|U|) with all zeros.

To calculate the similarity between user songs and all unique songs in the training data,

Calculate |U| of song (Si), Calculate |U| of song (Sj)

Now the intersection of the set of users who have listened to both songs, Si and Sj, is calculated, representing the number of users who have listened to both songs. The length of this intersection set is computed using the Python built-in function len (users_intersection).

$$L1 = S\_i \cap S\_j \quad (1)$$

Next is the union of the set of users who have listened to either Si or Sj, representing the total number of users who have heard to either or both songs. The length of this union set is computed using the Python built-in function len (users_union).

$$L2 = S\_i \cup S\_j \quad (2)$$

Finally, the Jaccard Index is calculated as the ratio of the number of users who have listened to both songs I and j (intersection set) to the number of users who have heard either or both songs (union set). The Jaccard Index is a measure of similarity between two sets and is defined as the ratio of the size of the intersection of two sets to the size of the union of the two sets.

$$C\_([S\_i][j]) = (len(L1)) \ / \ (len(L2)) \quad (3)$$

The Jaccard Index is assigned to the matrix's co-occurrence_matrix [Sj, Si] element, representing the similarity between song I and j. If there are no familiar listeners between songs I and j, then the Jaccard Index is assigned a value of 0.

The function begins by printing the number of non-zero values in the co-occurrence matrix. This indicates the density and significance of the relationships between songs based on user interactions.

Next, the function calculates the weighted average of the scores in the co-occurrence matrix for all songs. It does this by summing up the scores for each column in the matrix and dividing it by the total number of rows, representing the total number of songs in the dataset. The resulting array of scores is then converted into a list.

The list indices are sorted based on their values in descending order. The resulting indices are stored in a variable called "sort_index." This step ensures that the songs with the highest scores, indicating substantial similarity or popularity, appear at the beginning of the list.

Each song in the list is assigned a score based on its position. The higher the position, the higher the score. This scoring mechanism allows for ranking the user's songs in descending order, with the top-ranked songs being those with the highest scores. The function then recommends the top 20 songs based on this ranking, likely to be similar to the previously listened songs.

The function utilizes the co-occurrence matrix to identify similar songs based on user interactions. It calculates scores for each song and ranks them to recommend the top 20 songs to the user, considering their preferences and previously listened to songs.

V. EVALUATION

The evaluation of the model consists of two parts: the evaluation of the BERT model for semantic analysis (table I) and the evaluation of the Collaborative Filtering model for the recommendation system (table II). Both parts use the same evaluation metrics: accuracy, precision, recall, and F1 score.

To calculate these metrics, the true positive (TP), true negative (TN), false positive (FP), and false negative (FN) values are computed based on the recommended and expected outputs of the system.

The data is divided into training and testing datasets to compare the derived recommendation with the expected results.

The true positive (TP) occurs when the system recommends an item to a user, and the user likes and interacts with that item.

The true negative (TN) occurs when the system does not recommend an item to a user and the user does not interact with that item.

The false positive (FP) occurs when the system recommends an item to a user, but the user does not like or interact with that item.

The false negative (FN) occurs when the system does not recommend an item to a user, but the user likes and interacts with that item.

By comparing the projected values to the actual values in the test set, accuracy is the percentage of accurate predictions generated by the model. It is determined by dividing the total number of forecasts by the number of accurate guesses. [12]

$$Accuracy = ((TP + TN)) / ((TP + TN + FP + FN)) \qquad (4)$$

Precision quantifies the accuracy of the recommended items for all users. It is determined as the total number of recommended items divided by the number of relevant items [11].

$$Precision = TP / ((TP + FP)) \qquad (5)$$

Recall measures the percentage of relevant items the system has recommended. It is derived by dividing the total number of applicable items by the number of pertinent suggested things [4] .

$$Recall = TP / ((TP + FN)) \qquad (6)$$

The F1 score combines precision and recall into a single value and fairly assesses the system's performance. It is calculated as the harmonic mean of precision and recall, giving equal weight to both metrics [4].

$$F1\,Score = (2 * (precision * recall)) / ((precision + recall)) \qquad (7)$$

Evaluating these metrics can assess the performance of the BERT model for semantic analysis and the Collaborative Filtering model for the recommendation system.

TABLE I. EVALUATION RESULTS OF THE BERT MODEL

Parameters	Description	Values
Sample Size	Total number of sentences	110
True Positives	Number of positive sentences	33
True Negatives	Number of negative sentences predicted negatively by model	46
False Positives	Number of negative sentences but predicted positive or neutral by model	9
False Negatives	Number of positive sentences but predicted negative or neutral by model	4

Table I summarises the evaluation of a model's classification accuracy. A number of parameters are included to evaluate the model's accuracy. 110 sentences were utilized in this investigation, as indicated by the "Sample Size" field, which lists the number of sentences used for analysis. The number of positive statements the model adequately recognized is indicated by the term "True Positives." This evaluation shows that the model accurately identified 33 positive statements. The "True Negatives" number, however, shows how many negative statements the model correctly identified as such. The model properly recognized 46 negative statements in this case. The number of negative statements that the model misclassified as positive or neutral is shown by the "False Positives" parameter. In this evaluation, the model had 9 instances of misclassifying negative sentences. The number of positive statements the model mispredicted as negative or neutral utterances is represented by the "False Negatives" score. In this evaluation, the model had 4 instances where it failed to identify positive sentences correctly. Overall, this table provides a quantitative evaluation of the model's performance, highlighting the number of true positives and negatives and the number of false positives and negatives.

TABLE II: EVALUATION RESULTS OF COLLABORATIVE FILTERING MODEL

Parameters	Description	Values
Sample Size	Total number of recommendations	100
True Positives	Number of songs correctly recommended	85
True Negatives	Number of correctly non-recommended songs	0
False Positives	Number of songs not precisely recommended	15
False Negatives	Number of incorrectly non-recommended songs	0

Table II presents the evaluation results of a recommendation system's performance in suggesting songs. It comprises several parameters that assess the accuracy of the recommendations. The "Sample Size" indicates the total number of recommendations evaluated, which is 100. The "True Positives" value indicates the accurately recommended songs by the proposed system. In this evaluation, the recommendation system accurately suggested 85 deemed suitable songs.

On the contrary, the "True Negatives" value represents the quantity of songs that the system accurately detected as being unrecommended. However, in this evaluation, the system did not correctly classify any songs as not recommended, resulting in a value of 0 for true negatives. The number of songs that the system inaccurately recommended is indicated by the "False Positives" parameter. These are songs that the system suggested but were not suitable for the given criteria. In this analysis, the system had 15 instances where it provided recommendations that didn't comply with the desired criteria. The "False Negatives" value represents the number of songs incorrectly recommended by the system. Despite being appropriate, these songs met the requirements but weren't recommended by the algorithm. In this evaluation, the system did not have any instances of incorrectly not recommending songs, resulting in a value of 0 for false negatives.

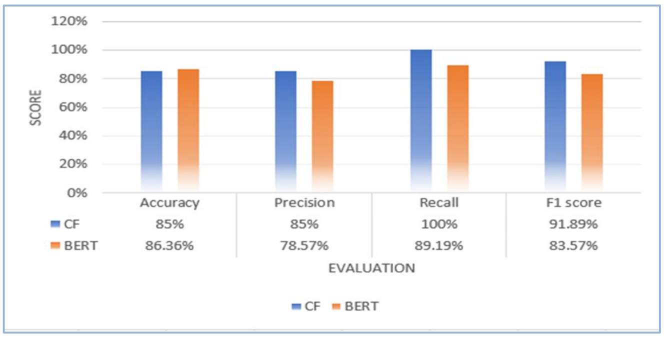

Figure 2: Results of BERT and Collaborative Filtering Models to check the Accuracy, Precision, Recall, and F1 Score of the proposed Recommendation System

Figure 2 presents the results of BERT and Collaborative Filtering models to check the proposed recommendation system's accuracy, precision, recall, and f1 score. The first evaluation is for the BERT model, a type of language model used for natural language processing. The assessment indicates that the model achieved an accuracy of 86.36%, meaning that it correctly classified 86.36% of the samples. The precision value is 78.57%, indicating that when the model predicted a positive result, it was correct 78.57% of the time. The recall value is 89.19%, which means the model correctly identified 89.19% of all positive samples. Finally, the F1 score is calculated to be 83.57%, representing the harmonic mean of precision and recall and providing a balanced measure of their performance. These results suggest that the BERT model performs reasonably well on the evaluated task.

The second evaluation is for collaborative filtering, a technique commonly used in recommender systems. The evaluation indicates an accuracy of 85.00%, indicating that the system accurately predicted 85.00% of the recommendations. The precision is also 85.00%, indicating that when the system recommended a product, it was correct 85.00% of the time. The recall value is 100.00%, meaning the system identified all relevant recommendations. The F1 score is calculated to be 91.89%, indicating that the system performs well in terms of both precision and recall. These results suggest that the collaborative filtering system effectively provides recommendations to users based on their preferences.

In evaluating the BERT model, the model's output is categorized into three main sentiments: positive, neutral, and negative. These sentiments are further divided into positive, slightly positive, neutral, negative, and slightly negative. The BERT model classifies the input text into these sentiment categories. These subdivisions are combined to form the three major categories of positive, negative, and neutral sentiments for evaluation purposes.

VI. Discussion

As for the BERT model, it achieves an accuracy of 85.5%, indicating that it correctly classifies about 86% of instances. While this is a good accuracy score, it is essential to consider other metrics to gain a more comprehensive understanding of the model's performance. The precision of the model is 78.6%, correctly identifying about 79% of the positive instances. The recall is 89.2%, meaning it correctly identifies approximately 89% of the optimistic scenarios. The F1 score, which represents the harmonic mean of recall and precision, is 83.6%. This suggests the model balances precision and recall but could still be improved.

The collaborative filtering model demonstrates high accuracy of 85% and precision of 85%, indicating its effectiveness in correctly classifying positive instances. A higher precision value typically leads to higher accuracy in the model's results. Furthermore, the recall score of 100% suggests that the model successfully captures all positive instances in the dataset. However, the lower F1 score of 0.459 suggests that the model could benefit from further improvements to achieve optimal performance. Evaluating the collaborative filtering and BERT models involves techniques such as accuracy, precision, recall, and F1 score to assess their performance. Further analysis may be necessary to determine whether the models' performance is well-rounded or whether enhancements can improve efficiency.

VII. Conclusion

In conclusion, this research paper has presented a personalized music recommendation system that combines collaborative filtering and semantic analysis techniques to provide accurate and relevant music recommendations. By leveraging the strengths of both approaches, the system successfully addresses the cold start problem for new users using semantic analysis implemented with the BERT model.

The BERT model classifies text based on moods categorized into positive, negative, and neutral. These moods are utilized for the initial recommendation, and subsequently, collaborative filtering is employed based on the songs that the user has chosen to generate further recommendations.

The collaborative filtering and BERT model evaluation utilized various evaluation parameters such as accuracy, precision, recall, and F1 score. These parameters enabled a comprehensive analysis of the model's performance. However, additional analysis may be required to ensure a balanced performance and identify potential areas for optimization.

The findings of this research demonstrate that the proposed recommender system, combining collaborative filtering and semantic analysis, holds promise in improving the accuracy and effectiveness of personalized music recommendations. Future work could focus on scalability and efficiency and explore further integrating other techniques like deep learning to enhance the system's performance.

Overall, this research contributes to the ongoing development of music recommendation systems by presenting a unique perspective and addressing the challenges associated with cold start issues. The proposed system has the potential to enhance users' music listening experiences by providing tailored recommendations and supporting music discovery.

References

[1] C. N. Dang, M. N. Moreno-García and F. De la Prieta, "Using Hybrid Deep Learning Models of Sentiment Analysis and Item Genres in Recommender Systems for Streaming Services," MDPI ELECTRONICS, p. 16, 2021.

[2] E. Liebman, M. Saar-Tsechansky and P. Stone, "THE RIGHT MUSIC AT THE RIGHT TIME: ADAPTIVE PERSONALIZED PLAYLISTS BASED," MIS Quarterly, vol. 43, p. 28, 2019.

[3] J. Bobadilla, F. Ortega, A. Hernando and J. Bernal, "A collaborative filtering approach to mitigate the new user cold start problem," Knowledge-Based Systems, p. 14, 2011.

[4] F. O. Isinkaye, Y. O. Folajimi and B. A. Ojokoh, "Recommendation systems: Principles, methods and evaluation," Egyptian Informatics Journal, p. 13, 2015.

[5] A. Roy and M. Ojha, "Twitter sentiment analysis using deep learning models," IEEE 17th India Council International Conference (INDICON) | 978-1-7281-6916-3/20/$31.00 ©2020 IEEE |, p. 6, 2020.

[6] K. Dominik and L. Emanuel, "Popularity Bias in Collaborative Filtering-Based Multimedia Recommender Systems," dominikkowald.info, p. 11, 2016.

[7] G. Tzanetakis and P. Cook, "Musical Genre Classification of Audio Signals," IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING, vol. 10, p. 10, 2002.
[8] X. Zhu, Y.-Y. Shi, . H.-G. Kim and K.-W. Eom, "An Integrated Music Recommendation System," EEE Transactions on Consumer Electronics, vol. 52, no. 9, p. 9, 2006.
[9] D. Sánchez-Moreno, M. N. Moreno-García, N. Sonboli, B. Mobasher and R. Burke, "Inferring User Expertise from Social Tagging in Music Recommender Systems for Streaming Services," Springer International Publishing AG, part of Springer Nature 2018, p. 11, 2018.
[10] K. Yoshii, M. Goto, K. Komatani, T. Ogata and H. G. Okuno, "Hybrid Collaborative and Content-based Music Recommendation Using Probabilistic Model with Latent User Preferences," SMIR, vol. 6, p. 6, 2006.
[11] V. S. Tida and S. Hsu, "Universal Spam Detection using Transfer Learning of BERT Model," arXiv:2202.03480, p. 9, 2022.
[12] J. Brownlee, "Failure of Classification Accuracy for Imbalanced Class Distributions," p. 17, 2020.
[13] P. K. Singh, M. Sinha, S. Das and P. Choudhury, "Enhancing recommendation accuracy of item-based collaborative filtering using Bhattacharyya coefficient and most similar item," Springer Science+Business Media, LLC, part of Springer Nature, p. 24, 2020.
[14] C. DHAHRI, . K. MATSUMOTO and K. HOASHI, "Mood-Aware Music Recommendation via Adaptive Song Embedding," IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), p. 4, 2018.
[15] Kamehkhosh, D. Jannach and L. Lerche, "Personalized Next-Track Music Recommendation with Multi-dimensional Long-Term Preference Signals," ceur-ws.org, p. 9, 2012.
[16] W. Daliang and G. Xiaowen, "Multidimensional Analysis of Music Education System Based on Multi-Intelligent Recommendation," Hindawi, vol. 2022, p. 8, 2022.
[17] C. Yixiao and L. Peng, "Personalized Music Hybrid Recommendation Algorithms Fusing Gene Features," Hindawi, vol. 2022, p. 8, 2022.
[18] Y. Lu, T. Koki, G. Chakraborty and M. Matsuhara, "Performance Comparison of Clustering Algorithm Based Collaborative Filtering Recommendation System," MISNC2020&IEMT2020: Proceedings of the 7th Multidisciplinary in International Social Networks Conference and The 3rd International Conference on Economics, Management and Technology, p. 6, 2020.
[19] Y. Sun, D. Xu, Y. Xie and W. Dai, "Intelligent Analysis of Music's Affective Features and Expressive Pattern," 2020 35th Youth Academic Annual Conference of Chinese Association of Automation (YAC) | 978-1-7281-7684-0/20/$31.00 ©2020 IEEE , p. 6, 2021.
[20] M. Bakhshizadeh, A. Moeini and M. Latifi, "Automated Mood Based Music Playlist Generation By Clustering The Audio Features," 2019 9th International Conference on Computer and Knowledge Engineering (ICCKE), p. 8, 2019.
[21] LEE, K. YOON, D. JANG, S.-J. J. JANG, S. SHIN and J.-H. KIM, "MUSIC RECOMMENDATION SYSTEM BASED ON GENRE DISTANCE AND USER PREFERENCE CLASSIFICATION," Journal of Theoretical and Applied Information Technology , vol. 96, p. 8, 2018.
[22] D. Wang, G. Xu and S. Deng, "Music Recommendation via Heterogeneous Information Graph Embedding," 978-1-5090-6182-2/17/$31.00 ©2017 IEEE, p. 8.
[23] D. Jannach, I. Kamehkhosh and L. Lerche, "Leveraging Multi-Dimensional User Models for Personalized Next-Track Music Recommendation," Association for Computing Machinery, p. 8, 2017.
[24] Y. Hu and M. Ogihara, "NEXTONE PLAYER: A MUSIC RECOMMENDATION SYSTEM BASED ON USER BEHAVIOR," 12th International Society for Music Information Retrieval Conference, p. 6, 2011.
[25] H. I. James, J. A. Arnold, J. M. M. Ruban, M. Tamilarasan and R. Saranya, "EMOTION BASED MUSIC RECOMMENDATION SYSTEM," International Research Journal of Engineering and Technology (IRJET), p. 6, 2019.
[26] M. Schedl, P. Knees, B. McFee, D. Bogdanov and . M. Kaminskas, Music Recommender Systems, TU Wien, Vienna University of Technology.
[27] H.-C. Chen and A. L. Chen, "A music recommendation system based on music data grouping and user interests," Association for Computing Machinery, p. 8, 2001.