iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 **Issue:** IV **Month of publication:** April 2023

DOI: https://doi.org/10.22214/ijraset.2023.50754

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue IV Apr 2023- Available at www.ijraset.com

Music Recommendation System Using Deep Learning

Raj Kumar Saw[1], Sumit Kumar[2], Nidhi Mishra[3]
[1, 2]*B.Tech (CSE), ASET, Amity University, Noida, UP, India*
[3]*Dept. of CSE, Amity University, Noida, UP, India*

Abstract: ***As part of this study, we present a music recommendation system that makes use of deep learning. The system learns a neural network that can anticipate the user's musical preferences and provide personalised playlist suggestions based on the user's listening habits. The proposed method considers both the user's explicit tastes and those that may be deduced from their listening patterns. The technology is able to adapt to a user's evolving preferences over time and provide more relevant recommendations. Software that predicts what a user will wish to buy based on their tastes and prior purchases is called a recommendation system. Although the emphasis of this paper is on improving music recommendation systems, the approach outlined here might be applied to a broad range of other platforms and domains as well, including video sharing sites like YouTube and Netflix as well as online retailers like Amazon. System efficiency decreases as complexity increases. Our technique, Tunes Recommendation System (T-RECSYS), provides an efficient recommendation system that can make predictions in real time by integrating data from both content-based and collaborative filtering into a deep learning classification model. By applying our strategy to the Spotify Recsys Challenge dataset, we find a threshold that provides an optimal trade-off between false positives and false negatives, increasing accuracy to 88%.***
Keywords: music recommendation system, deep learning, recommendation system, etc.

I. INTRODUCTION

It's no wonder that music recommendation algorithms have become more popular as more and more people use streaming services to listen to their favourite music. With the use of these techniques, users may personalise their journey towards music discovery. Often, music recommendation systems will use collaborative filtering and content-based approaches. Yet, the dynamic complexity of consumers' preferences is not captured well by these methods. A powerful technique for creating cutting-edge music recommendation systems, deep learning has arisen in recent years. In this study, we present a music recommendation system driven by deep learning that can make inferences about a user's likes from their listening history. The algorithm takes into account both explicit and implicit user preferences, altering the quality of its recommendations based on users' changing preferences. The purpose of recommendation systems is to provide users with personalised suggestions that perfectly match their preferences and needs [1]. Unappreciated despite the fact that they affect almost every aspect of our life, from choosing what to watch or listen to on the television to buying a product online. With so many potential applications, designing such systems necessitates a wide range of decisions and approaches. When it comes to giving suggestions, two dominant perspectives existed. Content-based filtering is a method of making product suggestions based on the degree to which items share certain characteristics [2]. If user A likes product B, then user B is more likely to also enjoy product A. Yet, collaborative filtering works because it uses users' shared interests in music, games, and merchandise to provide recommendations. Users A and B's shared traits allow us to make educated guesses about their shared tastes and preferences. It is also important to know whether a recommender system can operate in real time. [4]. It is said that real-time functionality calls for pre-processing steps to be carried out during data collecting. For this study, we create a novel algorithm for generating k-recommendations of music for users. The Tunes Recommendation System (T-RECSYS) returns the top k-scoring songs from a database by using a learned hybridization of content-based and collaborative filtering to give a score to each song in the database based on the user's preferences. Several crucial metadata aspects, such as genre, mood, and tempo, may be inferred from the user's input by mining their prior activities and preferences in the system. These considerations are included into our recommendation system. Based on these inputs, a deep learning classification model determines the likelihood that a user would like a piece of music. [8] [9]. As an example, consider the weekly playlists curated by music streaming services like Spotify based on a user's listening habits and interests. When the user is presented with suggested music to add to an existing playlist, T-RECSYS may be utilised for the same purpose. A similar challenge was set by Spotify in their 2018 RecSys Challenge, underlining the necessity for such research despite the fact that big recommendation systems have been installed on the web.

©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue IV Apr 2023- Available at www.ijraset.com

II. RELATED WORK

Solution proposals were published in articles that found similar problems with recommendation systems, such as a lack of real-time updates or the use of the wrong sorts of variables. In order to learn about the user's wants and requirements, Fang et al. [11] provide a technique that expands on a questionnaire. It is stated by the authors that "the data obtained from the survey is linked with musical qualities to narrow down the initial 384,500 songs to the 1000 songs the user is most likely to like" [11]. Next, we use machine learning clustering methods to analyse the questionnaire and divide the takers into like-minded musical taste subgroups [11, 12]. By integrating user profiles to generate group profiles, the programme employs collaborative filtering to determine which proposals are most likely to be accepted and suited by the whole group [11]. The focus of these recommendations is on group physical therapy sessions, particularly Zumba, yoga, and pilates. User Profiling (UP) [11] is a recommendation method that generates early results, which are subsequently refined using Reinforcement Learning (RL). In the questionnaire, you may give a score out of five for the music's tempo, loudness, energy, positivity, familiarity, and rhythmic dominance [11]. The capacity to rigorously stick to preferences that may be necessary in choice exercises is facilitated. While this research uses multi-variable, collaborative, and content-based filtering, it is only applicable to the context of making exercise music recommendations and is hence not useful in other contexts. Jiang et al. [[2017]] help recommendation algorithms rate music based on a variety of factors. Song comparisons were made using a recurrent neural networks (RNN) model [[13]]. While related systems have made reference to RNN models [14], our study is the first to quantitatively analyse the similarities across songs. Although musical elements were taken into account, no mention of real-time updates was mentioned in this presentation.

Content-based music information retrieval may be accomplished rapidly with the use of indirect matching, as proposed by Hayashi et al. [15]. (CBMIR). For this structure, we used the kinds of questions that would normally be asked during an offline investigation. They [15] used offline searches to quickly estimate similarities between documents found in online databases. The researchers used content-based filtering into their investigation, similar to methods employed in [13] and [12]. Collaborative filtering and automatic updates are two examples of functionality that this method was missing.

As suggested by Sunny et al. (2017) [9], real-time data streams may be analysed to deliver timely and accurate recommendations. Spark and other machine learning frameworks were used to develop adaptive recommendations in real time, which were then applied to the task of recommending TV channels. Mwinyi et al. [15] also presented a self-learning, predictive recommendation system for multimedia material that takes into account users' preferences before and after they make a selection, thus their work is comparable in that respect. While both approaches make use of real-time updates and a wide variety of factors, collaborative filtering is not one of them. In addition to these investigations, the Big Data Hub has also [13]

III. ALGORITHM

The method accepts a vector of data that may be used to both collaborative and content-based filtering. After being taught to recognise patterns in a user's listening habits, a deep neural network can determine with high certainty which songs to recommend.

A. Overview

The following is a high-level explanation of what T-RECSYS is. A deep learning model is fed data that represents nine songs the listener like, as well as a tenth song that symbolises a song the listener may or may not like. It gives back a number between zero and one that represents how likely it is that the user will like the tenth song based on their enjoyment of the previous nine. T-RECSYS finds the top k songs in terms of score and suggests them to the user.

B. Content-Based Filtering

T-ability RECSYS's capacity to gather data about songs enables content-based filtering of music. While analysing music, T-RECSYS considers several different factors, including genre, artist type, artist period, mood, tempo, and release year. The metadata values for 10 songs from a hypothetical playlist are shown in Table I. The pygn Python library allowed us to connect to the Gracenote API, one of several online APIs that provide access to this data. This information may be collected quickly and encodes the basic elements of a song. A deep neural network may be taught to use a content-based filtering approach once an input description has been provided. Just three of the six types of data may be put into arbitrary order (genre, artist type, and mood), although artist era, tempo, and release year are all flexible options. To feed the uncategorized data into the deep learning model, it is one-hot encoded, a machine learning technique in which a category variable is transformed into several boolean variables (which only takes numeric values). The table below shows the unique one-hot encodings that may be used to represent mood, artist type, and genre.

©IJRASET: All Rights are Reserved

C. Collaborative Filtering

T-RECSYS also includes collaborative filtering. In specifically, it takes the form of an index indicating how closely the tenth song is related to the user's nine favourite tunes. The degree to which two songs are similar is measured by how often they appear on the same playlist. It seems to reason that if two songs are regularly seen on the same playlist, then that must mean that one of them is recommending the other. There are two ways in which T- RECSYS uses this similarity score. One such metric is volume, which tracks how often a certain song pair appears on playlists. Nevertheless, this metric does not account for the frequency with which certain songs occur. It's possible that the frequency with which pairs of songs occur together is much lower than the frequency with which they appear separately. Because of this, T-RECSYS calculate the Sorenson index of every possible pair:

$$\frac{2|X \cap Y|}{|X| + |Y|}$$

where X represents a set of playlists containing the first song in the pair and Y represents a set of playlists containing the second song in the pair. The collaborative filtering input vector comprises 18 values since there are two metrics for each of the 9 possible song pairs.

Table 1: Example playlist when defined by song metadata

Playlist 1	Song 1	Song 2	Song 3	Song 4	Song 5	Song 6	Song 7	Song 8
Genre	Urban	Rock	Country	Pop	Rap	Pop	Classical	Hip-Hop
Artist Type	Female Group	Female Solo	Male Duo	Female Solo	Male Duo	Male Solo	Male Duo	Female Duo
Artist Era	2000s	1950s	1930s	2010s	1940s	1990s	1970s	1930s
Mood	Happy	Angry	Sad	Party	Sad	Energetic	Dance	Sad
Tempo	Fast	Fast	Medium	Slow	Medium	Fast	Medium	Medium
Release Year	2001	1962	1930	2009	1949	1986	1977	1932

Table 2: Example of one-hot encoding for different song metadata

Genre	Binary	Artist Type	Binary	Mood	Binary
Hip Hop	0000001000	Female Duos	000010000	Sad	0010000000
Country	0010000000	Male Duos	001000000	Sad	0010000000
Pop	1000000000	Female group	100000000	Happy	1000000000
Rap	0000100000	Male group	000100000	Sad	0010000000
Rock	0100000000	Female Solo	010000000	Angry	0100000000
Urban	0001000000	Female Solo	010000000	Party	0001000000

D. Deep Learning Model

The input vector is built after information is gathered using collaborative filtering and content-based methods. A deep neural network will be trained with this data to better understand the task of music recommendation. After a lot of trial and error with different hyperparameter settings, the tabled network structure specification was settled upon. Google's Tensorflow and the Keras deep learning framework in Python were helpful in the creation, training, and dissemination of our model. To create our training set, we modified Spotify's open playlist dataset [10]. For every playlist with 10 songs or more, we produced two training examples. In the first scenario, the user's enjoyment of the first nine songs in the playlist was used to make a recommendation about whether or not they would appreciate the tenth song (since the user who built the playlist liked all ten songs). As the previous nine tracks were used to decide if the user would like a random song, the second occurrence resulted in a negative recommendation. While it's impossible to predict whether or not a user would like a song selected at random, the chances are against it. When this was completed for every instance used in training, 20% were chosen at random to serve as a testing set.

Table 3: Neural network configuration

Layer	Parameters
Fully connected	Nodes: 70, Activation: reLU
Dropout	Dropout 80%
Fully connected	Nodes: 20, Activation: reLU
Dropout	Dropout 40%
Fully connected	Nodes: 7, Activation: reLU
Dropout	Dropout 20%
Fully connected	Nodes: 3, Activation: reLU
Dropout	Nodes: 1, Activation: Sigmoid

IV. RESULTS

Precision was selected as the primary criteria by which the model was evaluated:

$$Precision = \frac{t_p}{t_p + f_p},$$

where t_p represents true positives and fp represents false positives. Accuracy in this case refers to how frequently the suggested tune was liked by the listener. This is in line with our commonsense understanding of how such systems should function. The discrimination threshold is another key adjustable option. The discriminating threshold establishes the minimum value of the confidence score required to make a recommendation (as anticipated by the deep neural network). By raising the threshold, only songs with high confidence (i.e. high chance of being liked) are suggested. By default, a suggestion may be given when the confidence score surpasses 50% because it is expected that the listener would more likely than not enjoy the song.

Figures 1a-1d exhibit 4 different trials, showing accuracy achieved at varying levels of discrimination. As can be seen, using a 90% threshold led to 100% accuracy in three out of four trials, with 100% approval ratings for each and every music that was suggested. Precision scores that are outstanding even when utilizing the default 50% criterion are achieved in Trial 3, which achieved over 88% accuracy. There wasn't a huge amount of variation in the technique, since the sample standard deviation was just 8.58 per cent throughout the four trials.

V. DISCUSSION

All in all, we were really pleased with the accuracy we achieved. The configuration we have now does have a few drawbacks that should be mentioned. There was a fear that there wasn't enough reality behind negative suggestions. A user's musical tastes cannot be predicted from a set of playlists alone. Nevertheless, this is more of a dataset constraint than a flaw in T-RECSYS itself, since this is readily fixed using user data regarding disliked songs, which Spotify and other music services almost likely gather. Another issue is the method by which data for collaborative filtering is gathered. These calculations demand extensive, perhaps costly, worldwide knowledge of every song and playlist. However, using conventional indexing practices in a widespread rollout should make this shortcoming almost insignificant.

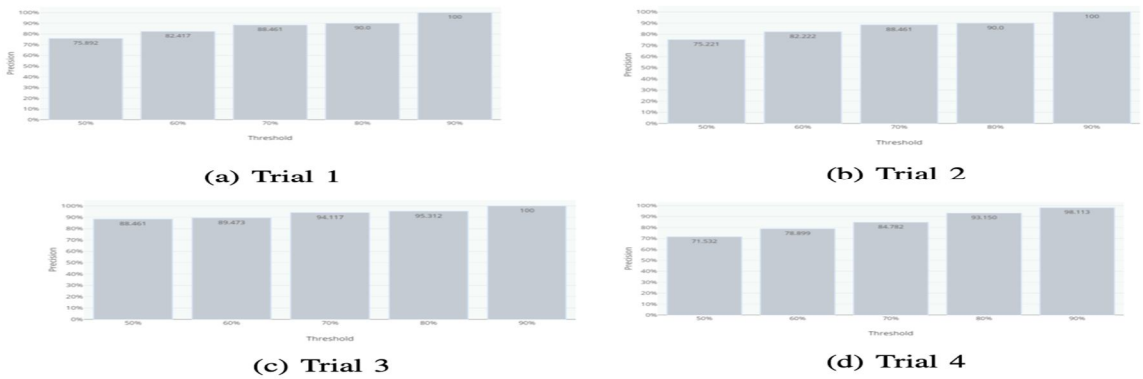


Fig. 1: Precision vs. Discrimination Threshold

There are obvious paths that might be explored in the future. When it comes to music recommendations, more songs than the nine utilised in this article might be employed to develop a preference baseline for listeners. More information, or even just the song's musical characteristics, might be utilised to describe each song. The effectiveness of both training and recommendations might be improved via the use of parallelization using Spark. The T-RECSYS framework might be expanded to include video, film, e-commerce, or any other area of interest. Each object, much like a song, contains properties that may be represented mathematically and hence encoded in the manner outlined in the article.

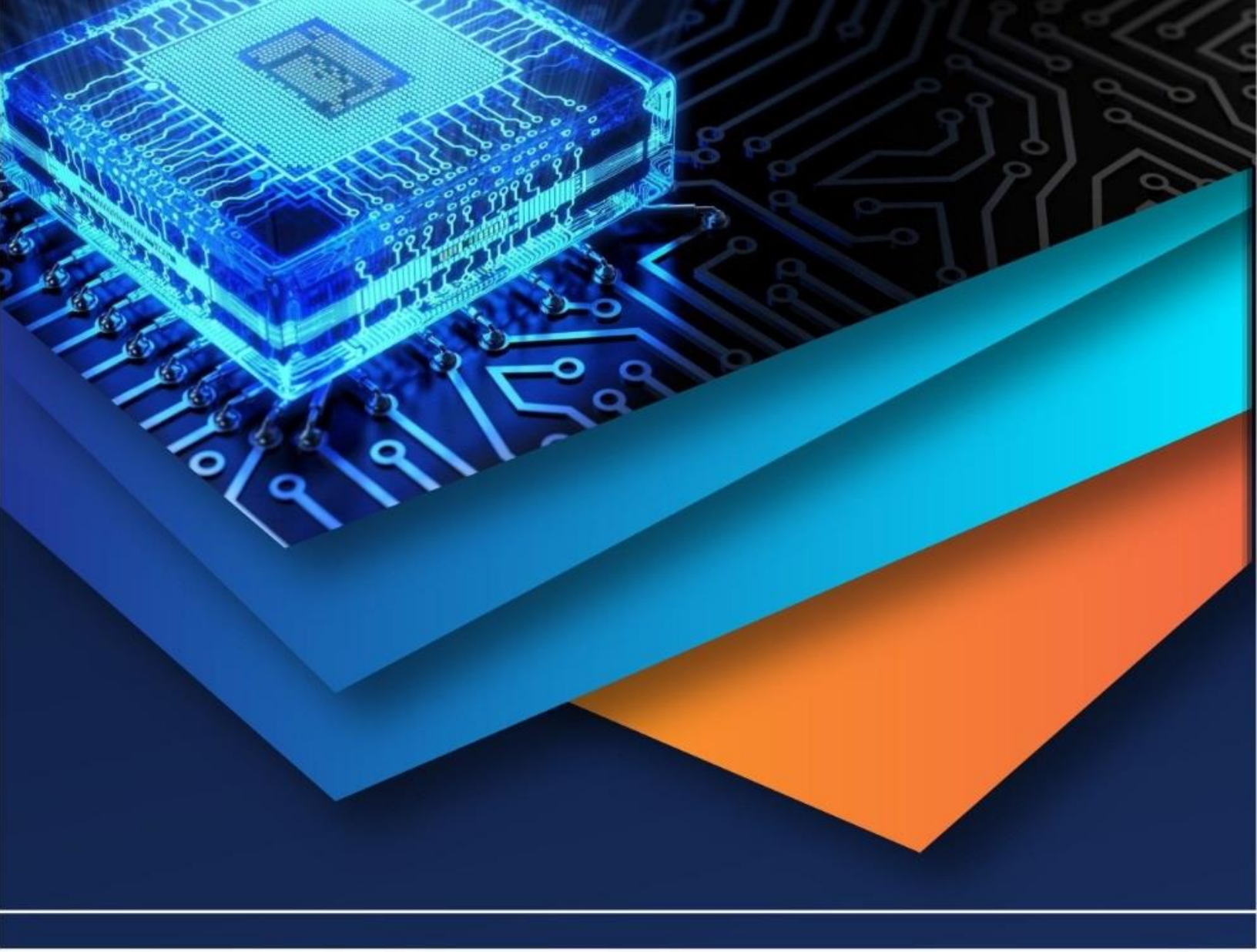
VI. CONCLUSION

Although we may not give much thought to recommendation algorithms, they greatly influence our everyday lives by deciding what we hear on Spotify and what we watch on YouTube. The complexity and accuracy of these systems are being continuously improved by researchers. In this research, we present T-RECSYS, a recommendation system that uses a deep learning model to learn user music tastes and provide song suggestions in the vein of services like Spotify, Pandora, and iTunes. We designed our algorithm to address issues that have plagued other algorithms in the literature, such as a lack of real-time updates and the ability to accept numerous kinds of variable inputs. The final result is a system with the potential for high suggestion accuracy and the flexibility to be applied to a wide variety of market services, including those offered by Amazon and Netflix.

REFERENCES

[1] P.Jomsri, S.Sanguansintukul, and W.Choochaiwattana,"Aframework for tag-based research paper recommender system: An ir approach," in 2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops, April 2010, pp. 103–108.
[2] M. Hassan and M. Hamada, "Improving prediction accuracy of multi- criteria recommender systems using adaptive genetic algorithms," in 2017 Intelligent Systems Conference (IntelliSys), Sept 2017, pp. 326– 330.
[3] Ruchika, A. V. Singh, and M. Sharma, "Building an effective rec- ommender system using machine learning based framework," in 2017 International Conference on Infocom Technologies and Unmanned Systems (Trends and Future Directions) (ICTUS), Dec 2017, pp. 215– 219.
[4] Y. Im, P. Prahladan, T. H. Kim, Y. G. Hong, and S. Ha, "Snn- cache: A practical machine learning-based caching system utilizing the inter-relationships of requests," in 2018 52nd Annual Conference on Information Sciences and Systems (CISS), March 2018, pp. 1–6.
[5] L. Shou-Qiang, Q. Ming, and X. Qing-Zhen, "Research and design of hybrid collaborative filtering algorithm scalability reform based on genetic algorithm optimization," in 2016 6th International Conference on Digital Home (ICDH), Dec 2016, pp. 175–179.
[6] I. Naser, R. Pagare, N. Wathap, and V. Pingale, "Hybrid music recommendation system: Enhanced collaborative filtering using context and interest based approach," in 2014 Annual IEEE India Conference (INDICON), Dec 2014, pp. 1–11.
[7] J. Lee, S. Shin, D. Jang, S. J. Jang, and K. Yoon, "Music rec- ommendation system based on usage history and automatic genre classification," in 2015 IEEE International Conference on Consumer Electronics (ICCE), Jan 2015, pp. 134–135.
[8] Y. Feng, Y. Zhuang, and Y. Pan, "Music information retrieval by detecting mood via computational media aesthetics," in Proceedings IEEE/WIC International Conference on Web Intelligence (WI 2003), Oct 2003, pp. 235–241.
[9] B. K. Sunny, P. S. Janardhanan, A. B. Francis, and R. Murali, "Implementation of a self-adaptive real time recommendation system using spark machine learning libraries," in 2017 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES), Aug 2017, pp. 1–7.
[10] C.-W. Chen, C. D. Boom, J. Garcia-Gathright, P. Lamere, J. McInerney, V. Murali, H. Rawlinson, S. Reddy, and R. Yon, "The million playlist dataset," Spotify - RecSys Challenge 2018, 2018. [Online]. Available: https://recsys-challenge.spotify.com/dataset
[11] J. Fang, D. Grunberg, S. Luit, and Y. Wang, "Development of a music recommendation system for motivating exercise," in 2017 International Conference on Orange Technologies (ICOT), Dec 2017, pp. 83–86.
[12] M. Ahmed, M. T. Imtiaz, and R. Khan, "Movie recommendation system using clustering and pattern recognition network," in 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), Jan 2018, pp. 143–147.
[13] M. Jiang, Z. Yang, and C. Zhao, "What to play next? a rnn-based music recommendation system," in 2017 51st Asilomar Conference on Signals, Systems, and Computers, Oct 2017, pp. 356–358.
[14] M. Kataoka, M. Kinouchi, and M. Hagiwara, "Music information retrieval system using complex-valued recurrent neural networks," in Systems, Man, and Cybernetics, 1998. 1998 IEEE International Conference on, vol. 5, Oct 1998, pp. 4290–4295 vol.5.
[15] T. Hayashi, N. Ishii, M. Ishimori, and K. Abe, "Stability improvement of indirect matching for music information retrieval," in 2015 IEEE International Symposium on Multimedia (ISM), Dec 2015, pp. 229– 232.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)