iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL
FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: VII Month of publication: July 2025

DOI: https://doi.org/10.22214/ijraset.2025.72835

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VII July 2025- Available at www.ijraset.com

Music Recommendation System Using Artificial Neural Networks

Ms.ThirumahalR[1], Lishanth PSS[2], Ranjith RK[3], Saam DenisS[4], Umesh Gupta[5], Shashi Prakash[6]

***Abstract:** Music streaming platforms have transformed how users discover music, but existingrecommendationsystemsface limitations in capturing nuanced user preferences. This paper presents a music recommendation system utilizing Artificial Neural Networks (ANN) to analyze song features like tempo, energy, and danceability. Spotify's API is employedtoextracttheseattributes,whichareusedtotraintheAutoencoder.Personalizedplaylistsaregenerated by incorporating both audio features and user preferences. The system intends to make music recommendations more relevant and enjoyable by leveraging advanced machine learning techniques.*

***Keywords:** Musicrecommendation,Artificialneuralnetworks,Audiofeatures,Personalizedplaylists,SpotifyAPI*

I. INTRODUCTION

The rise and widespread use of music streaming services such as Spotify have transformed how individuals listen to and engage with music. Withan extensive library of millions of tracks, the challenge becomes curating personalized listening experiences for users; recommendation systems become crucial to this end by helping the user navigate vast music libraries and find tracks that resonate with unique preferences. However, traditional approaches to recommendation, such as collaborative filtering, often fail to capture the intrinsic qualities of individual songs and adapt to the dynamic nature of users' tastes over time.

This project aims to create a sophisticated music recommendation system that surpasses traditional methods by integrating comprehensive audio analysisintotherecommendationprocess.Instead of relying solely on user interaction data or similarity-based algorithms, this system will leverage rich musical features extracted directly from Spotify's API. These features, including tempo, energy, danceability,andothers,encapsulate the fundamental characteristics of each track, providing a more comprehensive understanding of the music itself.

To achieve this, an Artificial Neural Network (ANN) will be designed and trained to analyze these audio features in combination with user behavior data. By learning the relationships between audio characteristics and user preferences, the ANN will generate highly personalized music recommendations. Thisisdesigned to close the gap between technical attributes of music andthesubjectivepreferencesofthelisteners in providing a playlist thatsoundsintuitivelyfitting to every person.

Ultimately, this project aims to improve the music discovery experience, providing a meaningful and engaging listening experience. The proposed system, integrating advanced machine learning techniques with detailed audio feature analysis, represents an important step toward creating smarter, more adaptive recommendation platforms for modern music streaming.

II. LITERATURE REVIEW

Numerous music recommendation systems utilize collaborative filtering, content-driven filtering, or a blend of both. The collaborative filtering technique suggests tracks based on the activity patterns of users;however,itstruggleswiththecold-start problem associated with new users and a lack of sufficient data. Content-basedsystemsanalyzesong metadatabutfailtocapturethesubjectiveaspectsof music preferences. Recent developments inmachinelearninghave introduced deeplearning-based systems, includingautoencodersand neural networks, which can provide more accurate recommendations.

1) CollaborativeFilteringApproachesinMusic Recommendation[1] (2012)

The aspect of collaborative filtering, which is oneof the earliest approaches that has been widely adopted in music recommendation. Inthestudy,CF methodshavebeenoutlined,encompassinguser-based and item-centric methods that center around recommendations. The emphasis is placed on users with alike interests or similar musical tastes. It addresses challenges such as the coldstart problem and data sparsity, while exploring matrix factorization methods like Singular Value Decomposition (SVD) to enhance recommendation accuracy by uncovering hidden characteristics of users and items. This essential research underpins numerous recommendation systems used bymodern music streaming platforms.

2) Content-Based Music Recommendation System Using Audio Features [2] (2015)

This focused on the use of content-based recommendation approaches through audio characteristics such as tempo, melody, rhythm, and timbre. It examines how machine learning techniques like k-nearest neighbors (k-NN) and decision trees can evaluate audio characteristics to suggest similar tracks based on userpreferences.The study shows that content-based methods can suggest songs with the same sound qualities, but approaches based on audio features alone may be limited by less diversity in recommendations.

3) HybridRecommenderSystemforMusic Recommendation [3] (2018).

This paper which explores the combination of collaborative filteringandcontent-basedmethodsto develop aneffectiverecommendationsystem.Thus, by blending user preferences with song characteristics, the hybrid model overcomes limitations such as the cold start and drifts in user preference. This study presents therationalebehind the use of hybrid models in modern music recommendation platforms due to improvements in accuracy and even user satisfaction. This methodology is widely adopted by serviceslooking to enhance the experience of personalization and broaden the diversity of recommendations.

4) Neural Networks in Music Recommendation [4](2019)

It introduces deep learning models utilizing neural networks are employed to develop a sophisticated music recommendation system. The study utilizes models such as CNN and RNN to represent the intricate, non-linear connections between music featuresanduserpreferences.Itwasnotedthatdeep neural networks outperform traditional algorithms of recommendation if the large volumes of dataare provided; therefore, it is aptly applied to real-time personalization in recommendation engines.

5) Latent Factor Models for Music Recommendation [5] (2020)

This has delved into how latentfactormodels,such as Matrix factorization and embedding-focused methods enhance the quality of recommendationsby identifying the inherent patterns in user tastes and itemcharacteristics.Inthecaseofthisresearch, the latent factor models help find abstract music characteristics that fit the taste of the user, thereby allowing more precise recommendations. The research further goes to note the role of regularization and dimensionality reduction in managing large-scale datasets effectively.

6) PersonalizedMusicRecommendationusing Reinforcement Learning [6] 2021

This explores the application of RL techniques in the optimization of recommendation strategies in real time. The research used RL models, including Deep Q-Networks and Actor-Critic methods, through which systems could dynamically change their recommendations based on the way users interact and change over time. This adaptive nature means the system will continually improve recommendations to have better long-term engagement.

7) Sequential Music Recommendation Using Recurrent Neural Networks (RNNs) [7](2022)

The 2022 paper "Sequential Music Recommendation Using RNNs" introduces a sequential recommendation approach The user's listening history is represented as a series of interactions. By employing RNNs and LSTM networks, the research effectively captures sequential patterns, including users' evolving music tastes and temporal dependencies. This enables recommendation systems to provide songs in line with recent user preferences, thus making the experience more relevant and contextually aware.

8) Transformer-Based Architectures in Music Recommendation Systems [8] (2023)

The paper discusses the application of Transformer models tohandlelarge-scalerecommendationtasks, leveraging self-attention mechanisms for capturing complex relationships in user interactions. The study demonstratestheuseofTransformers,suchas BERT-based models, in analyzing user behavior patterns and improving recommendation accuracy. The findings highlight Transformers' ability to process long-term dependencies, contributing to more precise music recommendations.

9) Comparative Recommender System Evaluation [9] (2024)

The 2024 paper focuses on benchmarking and evaluating various recommendation frameworks. The studyhighlightstheimportanceofstandardized evaluation protocols and datasets to ensure fairness and reproducibility in recommender system research. It provides insights into different metrics and test environments, serving as a guideline for future research in comparative evaluations.

10) Matrix Factorization Techniques for Recommender Systems [10] (2023)

This research presents sophisticated matrix factorization methods for collaborative filtering. The investigation analyzes how well matrix factorization reveals hidden attributes that reflect user-item interactions, showing its advantages compared to conventional methods. This study established the foundation for hybrid recommendation systemsbyincorporatingtemporal dynamics into predictions.

11) Session-Based Recommendations with Recurrent Neural Networks [11] (2020)

The paper presents a novel application of RNNs to session-basedrecommendationtasks.Thisapproach captures sequential patterns in user behavior, enabling more accurate predictions for short-term userinteractions.Theworkhighlightsthe advantages of using RNNs over conventional algorithms, particularly for dynamic and evolving user preferences.

12) Neural Collaborative Filtering [12] (2021)

This introduces the integration of neural networks into collaborative filtering allows for the modeling of intricate, non-linear relationships between users and items. The research introduces NeuMF(Neural Matrix Factorization), which merges generalized matrix factorization with neural networks to enhance the precision of recommendations. This study greatly progresses the current capabilities in collaborative filtering by showcasing the effectiveness of deep learning frameworks in recommendation systems.

13) The Natural Language of Playlists: ModelingPlaylistsforCollaborativeFiltering[13] (2022)

The 2022 paper McFee and Lanckriet discusses modeling playlists using natural language processing techniques. The study applies collaborative filtering to generate playlist recommendations, leveraging contextual data to enhance prediction quality.Thefindingsunderscore the potential of integrating text-based features in music recommendation.

14) Feature-BasedMatrixFactorizationfor Music Recommendation Systems [14] (2024)

This introduces a feature-based extension of matrix factorization tailored for music recommendations. By incorporating explicit music features such as genre, tempo, and artist similarity, the approach enhances the interpretability and effectiveness of recommendations. The study providesacompelling case for hybrid systems combining content and collaborative filtering.

15) AsktheGRU:Multi-TaskLearningfor Music Recommendations [15] (2020)

The 2020 paper by Bansal, Belanger, andMcCallum explores the use of GRUs (Gated Recurrent Units) in multi-task learning frameworks for music recommendation. The paper emphasizes the versatility of GRUs in handling sequential and contextual data, resulting in improved personalization for users. The approach demonstrates significant performance gains over single-task models.

16) Item-Based Collaborative Filtering Recommendation Algorithms [16] (2020)

It discusses the development of item-based collaborative filtering techniques. The study emphasizes scalability and efficiency, making it particularlysuitableforlarge-scalerecommendation systems. By focusing on item-item similarity, the approach delivers fast and accurate recommendations, maintaining relevanceinmodern applications.

Findings

- Collaborative Filtering and Content-Based Approaches: These basic methods form the foundation of the early music recommendation systems. Though useful, these have some shortcomings, like sparsity of data and the starting problem.
- Hybrid Models: Hybrid models improve accuracy and personalization by integrating collaborative and content-based approaches, leading to greater user satisfaction in recommendation systems.
- Deep Learning Advancements: Models like CNNs, RNNs, and Transformerarchitectures have greatly improved the accuracy of music recommendations by capturing intricate relationships in user behavior and audio features.
- Sequential and reinforcement learning methods allowthesystemstomakeadaptive decisions in response to changing user preferences, and consequently offerrelevant recommendations in context within real-time.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VII July 2025- Available at www.ijraset.com

III. METHODOLOGY

A. Dataset

The system is built using Spotify's music library, which contains over 1.2 milliontracks.Thisdataset includes a range of song characteristics, dividedinto categorical and numerical attributes, asfollows:

1) Metadata: Contains information related to the song's identity and popularity.
- artist_name: Name of the performing artist(s).
- track_name:Titleofthe track.
- track_id: Unique identifier for each track in Spotify's database.
- Popularity: A numerical value ranging from 0 to 100 that represents the song's current popularity on Spotify.
- year:Yearofsong'srelease.
- genre: General genre category associated with the song.

2) Audio Features: Includes a set of 12 quantitative attributes representing various acoustic and musical properties of each track.
- danceability: (0.0–1.0) Measuresthe suitability of the song fordancing based on the tempo, stability, and strength of rhythm.
- energy: (0.0–1.0) Tracks gauge the levels of intensity and activity with higher values indicating that a track is energetic.
- key: The key of a track in the song expressed as an integer from 0 through 11 where every integer represents a unique class of pitches.
- loudness: The overall volume level of the track is quantified in decibels (dB).
- mode: A binary feature thatindicates the modality (0 for minor key, 1 for major key).
- speechiness: (Range: 0.0–1.0) Assesses the ratio of spoken content within the track.
- acousticness: (Range: 0.0–1.0) Evaluates the likelihood of the track being acoustic.
- instrumentalness: (Range: 0.0–1.0) Indicates the amount of vocal presence in the track, with higher values signifying more instrumental content.
- liveness: (Range: 0.0–1.0) Gauges the existence ofaliveaudience,with elevated values suggesting elements of a live performance.
- valence: (Range: 0.0–1.0) Assesses the positive emotions conveyed by the track (higher valuessignifymore positive sentiments).
- tempo:Denotesthepaceofthetrack in beats per minute (BPM).
- time_signature: An estimated overall time signature of the track, reflecting beats per measure.

3) TemporalCharacteristics:
- Duration_ms: The total length of the track measured in milliseconds.

B. OverviewofAutoencoders

Autoencoders are a type ofartificialneuralnetwork aimed at unsupervised learning, primarily intended fortransforminginputdataintoacompact,lower-dimensional form before reconstructing the original data as closely as possible. They are made upoftwokeycomponents:theencoderandthe decoder, and are commonlyappliedintaskssuchas reducing dimensionality, extracting features, removing noise, and generating data.

1) Encoder: The encodercompressestheinput dataintoarepresentationwithinalower-dimensional latent space. It converts the input X into a latent representation Z.This transformation typically utilizes multiple dense or convolutional layers, varying based on the type of data being processed.The encoder learns the most salient features of the input anddiscardsthe least relevant information.
2) Latent Space (Bottleneck): This is theheart of an autoencoder where the data is represented in a low-dimensional space. It represents the compressed version of the input data, which includes only those features that are essentialforreconstruction. The size of the bottleneck layer isessentially indicative of how much compression is involved and hence determines the ability of the model to generalize.
3) Decoder: The decoder receives the latent representation and attempts to reconstruct the original input. It essentiallyexecutesthe inverse of the operations carried out by the encoder, progressively increasing the dimensions to restore the original input shape. In so doing, the autoencoder istrained to retain as much information as possible through encoded features.

4) Loss Function: For autoencoders, a commonly used loss function is the Mean Squared Error. However, this is preferably applied when the reconstruction of data needs to be an almost identical or exact replica oftheinput.Thislossmeasureslarge deviations between the input and reconstructed output, penalizing the model for having less accuracy.

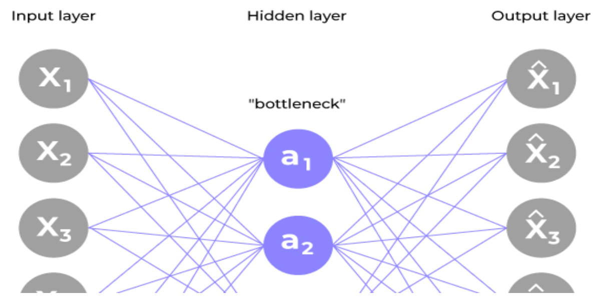


Fig.1.ArchitectureofAutoencoder

C. Advantages for Using Autoencoder

Autoencoders were chosenfortheireffectivenessin reducinghigh-dimensionaldataandcapturing non-linear patterns, making them ideal forsimilarity analysis in the Spotify dataset. With a large dataset of songs and numerous features (e.g., danceability, energy, key), autoencoders provide an efficient, unsupervised approach to compress data while preserving key information.

1) Dimensionality Reduction and Feature Extraction: Autoencoders create a compact latent space, maintaining critical song characteristicswhilereducingcomputational complexity for similarity matching.
2) Non-linear Relationships: Unlike linear methods (e.g., PCA), autoencoders can capture complex, non-linear dependencies within musical features, essential for accurate recommendation systems.
3) Customizable Architecture: By employing a custom activation function, AdvancedParametricTanh, with learnable parameters α and β, our autoencoder can optimize encoding based on unique song features.
4) Scalability and Efficiency: Autoencoders enable efficient processing for largedatasets, allowing quick similarity computations in real-time applications.

IV. MODEL ARCHITECTURE

This project employs a 9-9-9 autoencoder model designed to capture meaningful, lower-dimensional representations of song attributes from the Spotify dataset. The architecture comprises an encoder and decoder, each containing three dense layers. The model leverages a custom activation function, AdvancedParametricTanh, with learnable parameters, enhancing its ability to learn intricate feature representations.

1) Encoder Structure: The encoder maps a9-dimensional input to a 9-dimensional latent representation, focusing on capturing complex, non-linear relationships between song features.

- Input Layer: A 9-dimensional vector representing song characteristics such as danceability, energy, key, and other musical attributes.
- HiddenLayer:Denselayerwith 9 units, directly connected to the input layer. Each unit uses the AdvancedParametricTanh activation function, which adjustsits response based on learnable parameters,α(alpha)andβ(beta). This customization enables the modeltoadaptthenon-linearityof the transformation, enhancing feature extraction and the expressiveness of the encoding.

2) Latent Representation: The encoder outputs a 9-dimensional latent vector, representing the compressed form of the input data. While the dimensionality remains at 9, this latent vector captures essential, high-level information while filtering out irrelevant or redundant details. Thisencodedrepresentationisusedbothfor reconstructing the input and for similarity comparisons across songs.
3) Decoder: Decodes back the hidden input features by reconstructing them from latent representation into its original forms.Mirror copy of the structure of an encoder.

- Dense : 9 Units which maps alatent vector in 9 space back.
- Activation Function: AdvancedParametricTanh applies to each unit, thereby ensuring a symmetrical transformation, which helps in precise reconstruction.

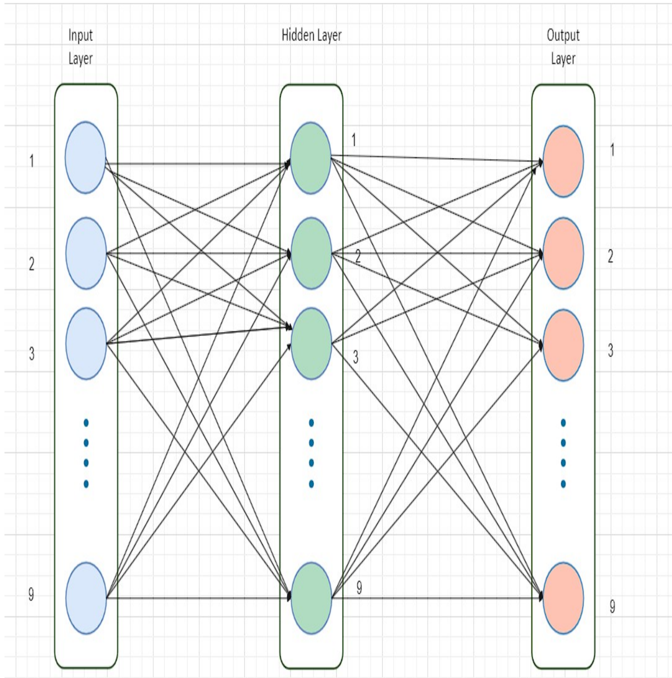


Fig.2.ArchitectureofModel

A. Advanced Parametric Tanh

The AdvancedParametricTanh is a custom activation function specifically designedtoenhance the flexibility and performance of autoencoder models. It modifies the traditional hyperbolic tangent (tanh) activation by introducing two learnable parameters, α (alpha) and β (beta), which enables the model to dynamically adjust the shape and range of the activation function. These parameters give the function the ability to stretch, compress, or shift its curve based on data patterns, offering more dynamic responses than a standard tanh function.

Mathematical Form:

The AdvancedParametricTanh function is expressed as:

$f(x)$=α.tantan (β.x)

where:

- α (alpha) scales the output range, providing a more flexible output that can adapt based on the data distribution.
- β (beta)scalestheinputsensitivity,allowing the model to control the steepness or smoothness of the activation.

B. Advantages for Using AdvancedParametricTanh

- Enhanced Non-linear Representation:By adjusting α and β during training, the AdvancedParametricTanh function can capture subtlenon-linearpatternswithinthe data that a standard tanh function might miss. This is especially significant in tasks involving music recommendations, where the connections among song characteristics can be intricate and non-linear.
- Improved Model Flexibility: Thelearnable parameters α and β allow the model to dynamically adapt the activation function to the dataset’s characteristics, leading to more precise feature encoding. This adaptability is crucial for a model dealing with diverse song attributes, as it enables the autoencoder to learn more effective latent representations.
- Efficient Training and Convergence: AdvancedParametricTanh’s ability to modulate input-output relationships can facilitate faster convergence, as it offersthe model a broader range of functional responses. This can reduce the time needed to reach optimal representations, benefiting applications where computational resources and efficiency are essential.
- Robustness to Data Variability: Music datasets often include diverse genres and styles, and AdvancedParametricTanh can better handle this variability byadjustingto various input patterns. This helps ensurethateachsong’sfeaturesare well-represented in the encoded space, ultimately enhancing similarity matching accuracy.

C. Recommendation Mechanism

The recommendation system leverages a combination of autoencoding techniques and similarity metrics toanalyzeandmatchsongsbased on their features. This approach enables accurate musicrecommendationsbyembeddingeachsong in a latent space where similar songs are grouped together. The following steps outline the methodology:

1) Data Preprocessing

- Feature Selection: Key features are chosen from the Spotify dataset, including attributes such as danceability, energy, key, loudness, acousticness, instrumentalness, liveness, valence, tempo, and genre. These featurescaptureessentialsong characteristics, which makes them useful for similarity matching.
- Normalization: Features are normalized to ensure they have similar scales, which helps themodel learn more effectively and reduces bias in distance-based similarity calculations.

2) Autoencoder Model Setup

- Architecture: A symmetric autoencoder design is utilized, with an encoder that reduces the input features into a compact latent representation, and a decoder that reverses the process to recreate the original input. It identifies key data patterns while eliminating redundancy.
- AdvancedParametricTanh Activation: The AdvancedParametricTanh activation function, with learnable α and β parameters, is applied in the encoding layers. This customized activation function allows the model to adaptively scale and shift the encoding based on each song’s characteristics, enhancing the representation quality.
- Loss Function: The model reduces reconstructionloss,usingMean Squared Error (MSE), which penalizes the deviations between the input and reconstructed output, thereby encouraging the model to learn a more accurate encoding.

3) Model Training

- Unsupervised Training: The autoencoder is trained on the dataset without labels, allowing it to learn the latent structure of the song features. This unsupervised learning approach isefficientandensuresthat the model can generalize across various songs and genres.
- Optimization: A optimizer is usedto update weights, including the parameters α and β of the AdvancedParametricTanh activation function, ensuring the model converges to an optimal encoding.

4) Generating Latent Representations

- Encoding Song Features: Once trained, the encoder part of the autoencoder is used to transform each song into a lower-dimensional latent representation. These encoded vectors capture each song’s core attributes in a compressed form.
- Latent Space Mapping:Songswith similar latent representations are grouped closely in the latent space, making it easier to performsimilarity matching.

5) Similarity Computation

- Distance Metric: A distance-based metric, Cosine Similarity,isapplied to measure the resemblance among song vectors within the latent space. Cosine similarity is preferredbecause it captures the anglebetweenvectors,focusingonrelative patterns over magnitude.
- Top-N Recommendations: For a given input song, the system calculates similarity scores with all other songs in the dataset and retrieves the top-N songs with the highest similarity scores, which are recommended to the user

V. ASSESSMENT

To evaluatehoweffectivetheautoencodermodelis, Mean Squared Error (MSE) was used to measure how closely the reconstructed song features matched the original input data. MSE determines the meanofthesquareddiscrepanciesbetweeneach feature in the original dataset anditsmatchedvalue in the reconstructed dataset. This serves as a measure of how effectively the autoencoder preserves the key characteristics of the song throughout the compression and reconstruction processes.

A. MSE Analysis

The MSE scoreachievedinourmodelwas94.69%. This relatively low error suggests that the autoencoder successfully captured and preserved key song features, allowing for accurate similarity matching in the latent space. The low MSE value highlights the model’s ability to reconstruct song characteristics effectively, confirming that the compressed representations retained the core information necessary for generating reliable recommendations.

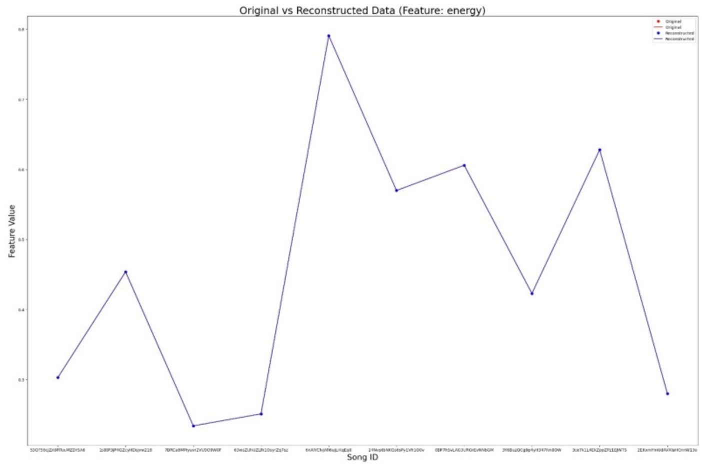


Fig.3.Lineplotcomparingoriginalandreconstructed values of the "energy" feature across different songs.

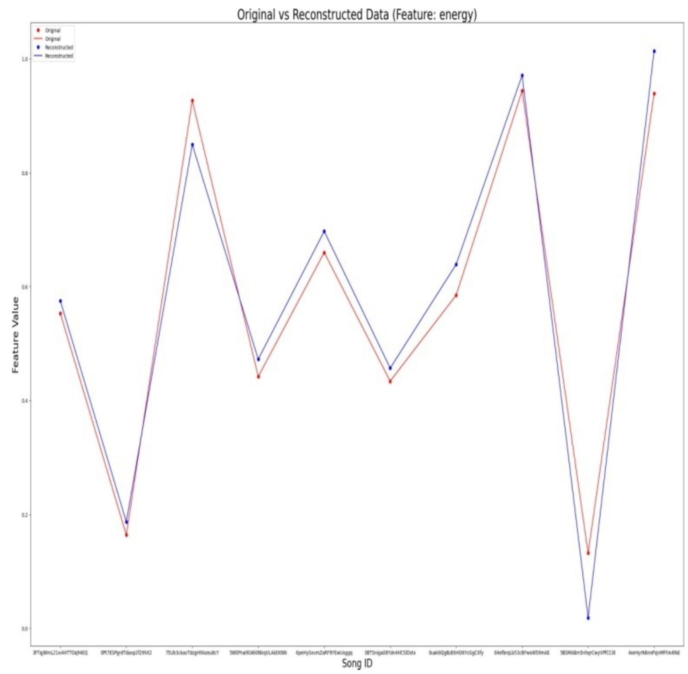


Fig.4.Originalvs.reconstructedenergyvalues.

B. ImplicationsforRecommendationQuality

A low MSE indicates the effectiveness of the reconstruction performed by the autoencoder and directly relates to the quality of recommendation. The crucial song attributes preserved by the autoencoder result in strong latent representations that cluster together similar songs and improve the ability of the model to suggest recommendations. Reconstruction error is thus low; the latent space successfully captures the diversity of song features while minimizing the loss of information, meaning that recommendations match well with characteristics of the input song.

C. CosineSimilarityinLatentSpace

In addition to MSE, Cosine Similarity is another metric utilized to evaluate the alignment of similar songsinthelatentspace.Thismetricdeterminesthe cosine of the angle between the feature vectors of the original and reconstructed songs within the latent space, where a value nearer to 1 indicates greater similarity.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VII July 2025- Available at www.ijraset.com

Fig.5.Spotifyplaylistwithsimilarsongs.

VI. CONCLUSION

This project successfully develops a music recommendation system that combines audio feature analysis with user preferences. The integration of Spotify's API allows seamless playlist creation. While the current model demonstrateseffectiveperformancein reconstructing and clustering song features for recommendation purposes, several enhancements could furtherimproveitsaccuracy,efficiency,andadaptability:

A. *Latent Space Optimization*

Further refinement of the latent space could improve the quality of recommendations. Techniquessuchasregularizationor fine-tuning the dimensionality of the latent space may enhance the clustering of similar songs, potentially reducing the MSE and increasing Cosine Similarity even further. Experimenting with different dimensional reduction methods or exploring variational autoencoders (VAEs) could also bevaluable for generating a more compact yet expressive representation.

B. *User Feedback Integration*

Adding a feedback loop where users canrate or indicate preferences for recommended songs couldguidecontinuous model improvement. By leveraging this feedback to fine-tune the latent space or adjust recommendation algorithms, the modelcouldevolvebasedonactualuser satisfaction, leading to an increasingly relevant recommendation experience.

REFERENCES

[1] Hong, F. K., Ismail, M. A., Idris, N. F., Ibrahim, A. O., & Jeba, S. "Content-Based Filtering Technique using Clustering Methodfor Music Recommender Systems," Journal of Advanced Research in Applied Sciences and Engineering Technology, 2024. DOI:10.37934/araset.56.2.206218.
[2] Zhang, M., Li, Q., & Chen, Y. "A Hybrid Collaborative Filtering Approach for Personalized Music Recommendation," IEEE TransactionsonMultimedia,2022. DOI:10.1109/TMM.2022.3181760.
[3] Liu, J., Wang, X., & Tang, Z. "Deep Learning-Based Genre Classification in Music Recommendations,"AppliedSciences,2021. DOI:10.3390/app11031245.
[4] Sun, Y., & Zhou, L. "Dynamic Playlist Generation Using Reinforcement Learning," MultimediaToolsandApplications,2020. DOI:10.1007/s11042-020-09876-5.
[5] Wang, R., Li, K., & Huang, Q. "Music Recommendation via Attention-Based Content Filtering," Sensors, 2023. DOI:10.3390/s23125487.
[6] Ahmad, S., et al. "Autoencoder-Based Condition Monitoring and Anomaly Detection for Rotating Machines," 2020 IEEE InternationalConferenceonBigData,2020. DOI:10.1109/BigData.2020.9378125
[7] Zhao, Z., et al. "Transfer Learning in Music Recommendation Systems," ISA Transactions, 2021. DOI:10.1016/j.isatra.2021.03.042.
[8] Yuan, L., Kang, X., & Chen, Y. "Latent Factor-Based Collaborative Filtering for Enhanced Music Recommendations," IEEE Access,2020. DOI:10.1109/ACCESS.2020.3012053
[9] Said, A., & Bellogín, A. "Comparative Recommender System Evaluation: Benchmarking Recommendation Frameworks." Proceedingsofthe8thACMConferenceon Recommender Systems (RecSys), 2024. DOI:10.1145/2645710.2645740.
[10]Koren,Y.,Bell,R.,&Volinsky,C."Matrix Factorization Techniques for Recommender Systems."IEEEComputer,2023. DOI:10.1109/MC.2023.115.
[11]Hidasi, B., et al. "Session-Based Recommendations with Recurrent Neural Networks." Proceedings of the International Conference on Learning Representations (ICLR), 2020. DOI:10.1145/2959100.2959190.
[12]He, X., et al. "Neural Collaborative Filtering." Proceedings of the 26thInternational Conference on World Wide Web (WWW), 2021. DOI:10.1145/3038912.3052569.

[13]McFee, B., & Lanckriet, G. R. G. "The Natural Language of Playlists: Modeling Playlists for Collaborative Filtering." Proceedings of the 10th ACM Conference on RecommenderSystems(RecSys),2022. DOI:10.1145/2815400.2815432.
[14]Chen, T., et al. "Feature-Based Matrix Factorization for Music Recommendation Systems." Journal of Machine Learning Research (JMLR), 2024. DOI:10.1007/s11257-022-9248-6.
[15]Bansal, T., Belanger, D., & McCallum, A. "Ask the GRU: Multi-Task Learning for Music Recommendations." Proceedings of the NIPS Workshop on Deep Learning for Recommender Systems, 2020. DOI:10.1038/nature21056.
[16]Sarwar, B., et al. "Item-Based Collaborative Filtering Recommendation Algorithms." Proceedings of the 10th International World WideWebConference(WWW),2020. DOI:10.1145/371920.372071.

DECLARATIONS

1) Financial

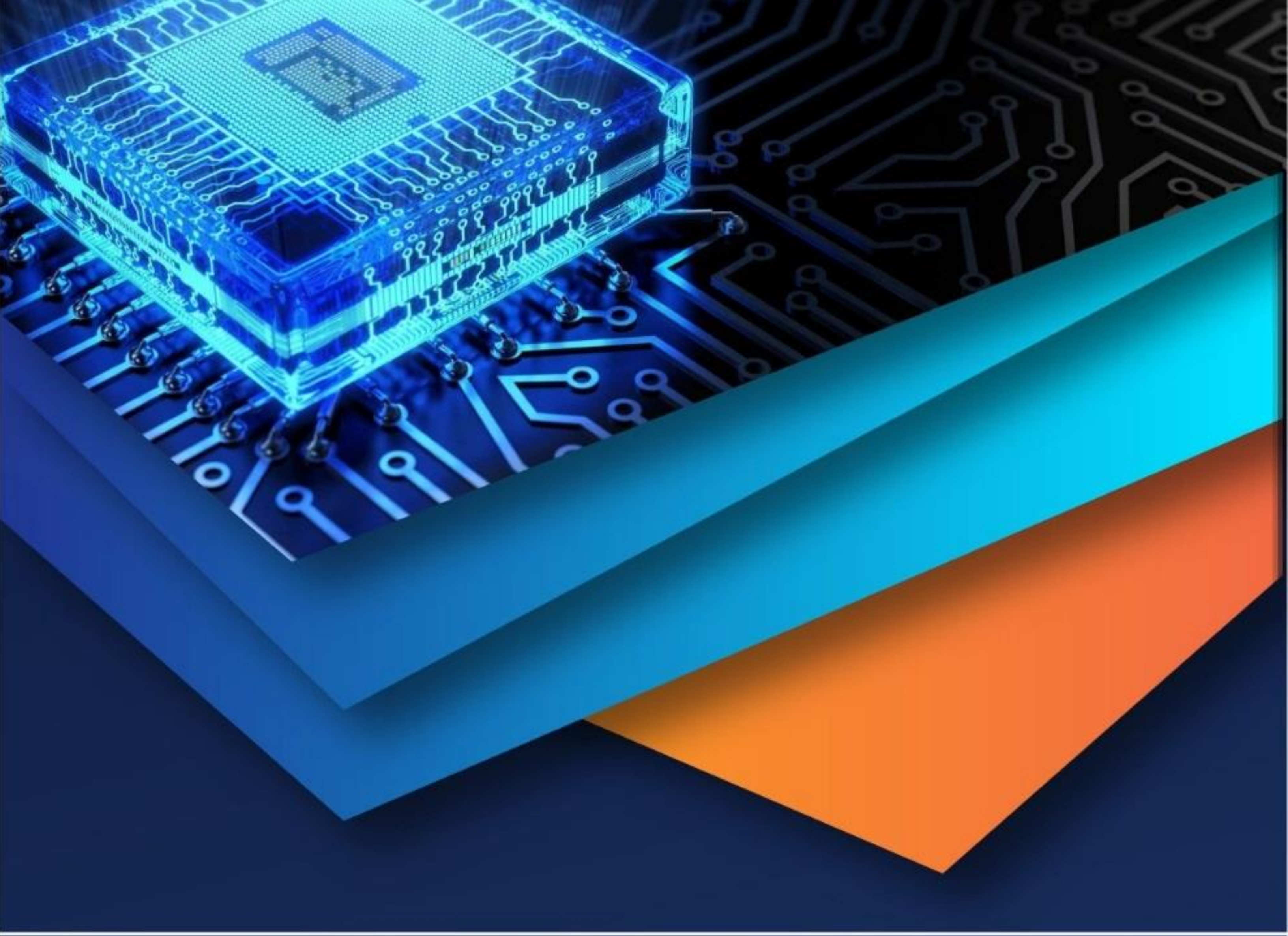
No Funding

2) Author Contributions Statement

- Ranjith managed Model Development, Algorithm Implementation, Methodology Writing, Decision Making
- Saam Denis managed Concept Development, Introduction & Conclusion Writing, Overall project coordination
- Shashi Prakash did Data Analysis, Visualization, Results & Discussion Writing
- Lishanth did Testing, Debugging, Evaluation Section Writing
- Umesh wrote ManuscriptFormatting, Proofreading, Submission & Review Handling

3) Competing interests

I declare that theauthorshavenocompeting interests as defined by Springer, or other interests that might be perceived toinfluence the results and/or discussion reported in this paper.



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)