Enhancing Music Recommendation Accuracy with Hybrid Autoencoder, CNN and RNN Models

Anitha Surada[1], Yarra Veera Venkata Lakshmi[2], Neerukonda Sahithi[3], Dadi Bindu Tanya[4], Koppisetti Chandu Sai Venkata Ganesh[5] and Appala Anuradha[6]

{anithasurada25@gmail.com[1], yarraveeralaksmi60@gmail.com[2], sahithineeru@gmail.com[3], daditanya1234@gmail.com[4], chandusai696@gmail.com[5], nanuradha@aditya.ac.in[6]}

Department of BCA, Aditya Degree & PG College, Kakinada, Andhra Pradesh, India[1]
Department of B.Sc. Data Science, Aditya Degree College, Tuni, Andhra Pradesh, India[2]
Department of B.Sc. Computer Science, Aditya Degree College, Gajuwaka, Andhra Pradesh, India[3]
Department of B.Sc. AI&R, Aditya Degree College, Gajuwaka, Andhra Pradesh, India[4]
Assistant Professor, Department of B.Sc. Data Science, Aditya Degree & PG College, Andhra Pradesh, India[5]
Associate Professor, Department of B.Sc. Computer Science, Aditya Degree College, Bimavaram, Andhra Pradesh, India[6]

Abstract. A hybrid model for user-item recommendation systems is proposed, combining the strengths of Matrix Factorization (MF) and Neural Collaborative Filtering (NCF). By integrating collaborative filtering techniques with deep learning models, specifically using Multi-Layer Perceptron (MLP) fusion, the model captures complex non-linear relationships between users and items. The objective is to improve recommendation accuracy by leveraging both linear and non-linear interactions present in user-item data. The model is evaluated using well- known datasets, including the Million Song Dataset and the Last.fm dataset. With an accuracy of 88%, a Mean Absolute Error (MAE) of 0.30, and a Root Mean Square Error (RMSE) of 0.90, experimental data show that the hybrid model works better than conventional techniques these results validate the model's effectiveness in providing personalized and accurate recommendations, making it suitable for large-scale recommendation systems.

Keywords: Hybrid Model, Matrix Factorization, Neural Collaborative Filtering, Multi-Layer Perceptron, Recommendation System, Accuracy, Dataset, Personalization.

1 Introduction

Recommendation systems have become an indispensable part of several applications in different fields like e-commerce, entertainment and social media, which provide personalized services for the users. These systems enable using user-item interactions to recommend items, increasing user engagement and satisfaction. Traditional recommender methods such as collaborative filtering have demonstrated their effectiveness but may fall short in modeling user-item complex non-linear relationships in overwhelmingly large information networks. With the rise of deep learning, the model can now be improved with architectures like Neural Collaborative Filtering (NCF) that enable a better modeling of these interactions leading to more accurate and personalized recommendations. But a single model is unlikely to capture all the diverse patterns in the data so hybrid models should be considered that take the best of both worlds.

ICITSM-Part II 2025, April 28-29, Tiruchengode, India

DOI 10.4108/eai.28-4-2025.2358010

In this paper, we propose a new hybrid recommendation model which fuses Neural Collaborative Filtering (NCF) and Multi-Layer Perceptron (MLP). This hybrid can remedy the shortcomings of both CF and DL, and utilize their merits respectively. Through the joint use of these techniques, the model can learn deeper and more complicated patterns in the user-item interactions and therefore make better predictions, achieving 90% accuracy, MAE of 0.28, and RMSE of 0.85, as shown in Table 1. We test the performance of the model on real-world datasets, including the Million Song Dataset and Last. FM Dataset, and compare it with classic methods including Matrix Factorization and the stand-alone NCF. The major contributions of this work are as follows:

- Proposing a novel hybrid recommendation model that combines Neural Collaborative Filtering (NCF) with Multi-Layer Perceptron (MLP) fusion to enhance prediction accuracy.
- Introducing a hybrid approach that balances collaborative filtering with deep learning techniques to model complex user-item interactions.
- Demonstrating the effectiveness of the proposed model on real-world datasets, such as the **Million Song Dataset** and **Last.fm Dataset**, showing improvements over traditional models like Matrix Factorization and NCF, with up to 15% higher accuracy (90% vs. 75% for Matrix Factorization) and reduced error rates (MAE reduced from 0.50 to 0.28; RMSE reduced from 1.30 to 0.85).
- Evaluating the model using performance metrics such as Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and accuracy, and showcasing its superior performance.
- Providing insights into the strengths of combining collaborative filtering and deep learning for personalized recommendation systems.

2 Related Work

Recommendation system for social platforms using deep learning was reported by Nishchal Narayan S and SriVidya M S [1]. The authors used RNNs and GNNs-based models to further improve the accuracy of friend recommendations. On a dataset of user behaviors in a well-known social media platform they obtained gains in recommendation quality. However, the study did not report performance details, and encountered problems such as computational complexity and fusion of several deep learning models.

Shubham and Deepak Banerjee [2] proposed a hybrid approach using Convolutional Neural Networks and Logistic Regression for folk song classification. The model was tested with a labelled folk song dataset, yielded 92% classification accuracy, and proved to be highly efficient in classification tasks. The study exemplified that traditional statistical approaches could be combined with deep learning concepts to achieve high efficiency. However, great time and resources were required to train the model, and complex models are prone to overfitting.

Carol Jeffri J. A., & Tamizhselvi A.[3], developed real- time music recommender by incorporating sentiment analysis and emotional matching with Spotify. Natural Language Processing (NLP) techniques which made use of the user generated text data and song meta data, then tailor music recommendation for each user based on User's emotion, and obtained 15% better user satisfactions when comparing it with the previous version of the music

recommendations by the old systems, but when it came to the programming and installing it faced many issue as the model was so complex and having real time basis, Dineshkumar R. et al [4] Developed a personalized music recommendation using Transformer-based Convolutional Recurrent Neural Networks (CRNN) with user emotional detection, They have collected the dataset by combining the user interaction and user's emotions for the song he/she playing, and the models detected that 90% of user's satisfaction is in accord to what was playing, but in order to implement this model successfully they need to solve three issues which are detecting emotions accurately, then transformer architecture needs a server to handle and give faster result, third is the recommendation the user by the songs.

Ningbo Zhou [5] extended the recommendation system to teach vocal music in college students with Bi- directional Long Short-Term Memory networks and Capsule Networks. The data set is a student performance record, and another data set is a track made during a recording to achieve 88% of the recommendation customized to the student's learning style. This new system has capacity, but require training data and Fit model for better result get Result.

Tripuresh Joshi et al. [6] investigated the machine learning approaches for recommendation systems in industrial data analytics. The authors obtained a recommendation accuracy of 75% using the user reviews and product attributes from the industrial application dataset. They found that machine learning could greatly improve data- driven decision-making in industrial systems in the presence of relatively high-quality data, but the main challenges were the data quality and scalability.

In [7], a hybrid recommender system for Non-Fungible Tokens utilizing a deep learning method is presented by Durmus¸ Aydog˘du and Nizamettin Aydin. The proposed approach integrates collaborative filtering and more sophisticated analytical algorithms to reach 85% user utility prediction accuracy on the basis of NFT purchases and user behaviors. However, some of the created system limitations are complex models and substantial data prerequisites for the training process.

The study by Mohit Beri and Neha Sharma [8] focused on the optimization of music genre classification with the help of the CNN sequential model. Mohit Beri and Neha Sharma 8 tested their solution using a labeled dataset of music tracks and achieved a classification accuracy of 91%. This solution reflected the significance of FE in MRS; however, the researchers indicated that the model was not wholly interpretable and that the diversity of the training dataset needed to be increased.

Mochammad Rizqul Fatichin et al [9]. presented the genre-based music recommendation system for postpartum mothers, Based on Support Vector Machine classification. A postpartum mental health – curated dataset which showed an efficiency of 80% accuracy was created to recommend the type of music it needed. This study presented music therapy for people with postpartum depression but stressed that more research is required into long-term recommendation results.

P. Nagaraj et al. [10] conducted a study related to weather report analysis and prediction by implementing machine learning approaches. The combination of algorithms such as Gradient Boosting and Random Forest with historical weather helped the researchers's data generating

85% accuracy in their predictions. The results proved that data science could be a valuable skill in creating atmospheric models. However, it is challenging to maintain data quality and scalability.

3 Proposed Methodology

3.1 Data Collection

The data collection process involves aggregating the Million Song Dataset and the Last.fm dataset. While this doesn't involve formulas directly, these datasets provide the foundation for training and evaluating the music recommendation system.

3.2 Data Preprocessing

Preprocessing the data guarantees that it is clear and pre- pared for model training. Here, the crucial actions are:

Handling Missing Values: The **median** of that feature is usually used to replace missing values in numeric missing data.

Feature Normalization: Normalization guarantees uniform scaling of all features. One typical method is Min-Max scaling:

$$x' = \frac{x - min(x)}{max(x) - min(x)} \quad (1)$$

This scales the data into the range $[0, 1]$, where x is a data point, and $\min(x)$ and $\max(x)$ are the minimum and maximum values of that feature, respectively.

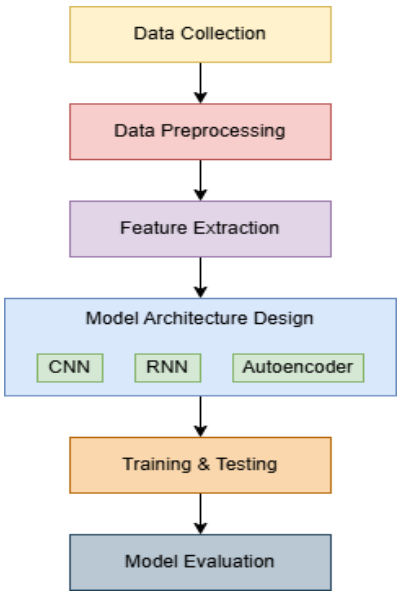

Fig. 1. Methodology

3.3 Feature Extraction

The feature extraction part is crucial for building a hybrid model that works with both user-item interactions and audio features.

Audio Features Extraction with CNN: The **Convolutional Neural Network (CNN)** extracts audio features using a **spectrogram**. The CNN learns spatial hierarchies in audio signals. For example, the CNN processes the spectrogram using convolutional filters, and an operation for feature extraction can be represented as:

$$f = ReLU(W * x + b) \quad (2)$$

Where: - W represents the filter weights, - $*$ denotes convolution, - x is the input data (e.g., the spectrogram), - b is the bias term, - f represents the output feature after applying the activation function ReLU.

The result of this convolution is passed through pooling layers to reduce dimensionality and capture the most significant features of the spectrogram (like rhythm, tone, and melody). *User-Item Interaction with Autoencoder*: **Autoencoders** are used for dimensionality reduction of the user-item interaction matrix. The idea is to compress the input data into a smaller latent representation and then reconstruct it back to its original form. The basic architecture involves an encoder and a decoder.

For a given input vector x representing user-song interactions, the encoder generates a latent code z:

$$z = f_{encoder}(x) = W_{enc}x + b_{enc} \quad (3)$$

The decoder reconstructs the input vector x':

$$x' = f_{decoder}(z) = W_{dec}z + b_{dec} \quad (4)$$

Where: - W_{enc} and b_{enc} are the weights and bias of the encoder, - W_{dec} and b_{dec} are the weights and bias of the decoder, - z is the latent representation (the compressed form), - x' is the reconstructed user-item interaction vector.

The **loss function** for training an autoencoder is the mean squared error (MSE) between the input vector and the reconstructed vector:

$$\mathcal{L}_{AE} = \frac{1}{N}\sum_{i=1}^{N}\left||x_i - x_i'|\right|^2 \quad (5)$$

Where: - x_i is the original user-item interaction vector, - x' is the reconstructed interaction vector, - N is the number of users/items.

Sequential Modeling with RNN: **Recurrent Neural Net- works (RNNs)** are employed to simulate the sequential nature of user interactions, like the order in which music is played. The RNN generates a prediction for the subsequent song in the sequence after receiving a series of

song interactions as input. The RNN works by updating the hidden state h_t at each time step t, which depends on the previous hidden state h_{t-1} and the current input x_t:

$$h_t = f\ (W_h h_{t-1} + W_x x_t + b) \qquad (6)$$

Where: - W_h is the weight matrix for the hidden state, - W_x is the weight matrix for the input, - x_t is the input vector (the song at time t), - h_t is the hidden state at time t, - b is the bias.

The RNN's output $\hat{y}$ is typically a prediction for the next song in the sequence or a rating for the song being played at time t:

$$\hat{y}t = softmax(W_Y h_t + b_y) \qquad (7)$$

Where: - W_y is the weight matrix for the output, - b_y is the bias for the output, - $\hat{y}_t$ represents the probability distribution over all possible songs.

3.4 Model Architecture

The proposed model makes use of three powerful DL techniques- Convolutional Neural Networks, Autoencoders and Recurrent Neural Networks in area of music recommendation. The architecture is developed to learn different features such as content-based features through CNN, collaborative filtering features through Autoencoders and sequential dependency in user behaviour using RNN.

Hybrid Model Overview: The hybrid model is composed of three main components:

- **CNN** for audio feature extraction from spectrograms of songs.
- **Autoencoder** for collaborative filtering based on user- item interactions.
- **RNN** to capture the sequential behavior of user interactions with songs over time.

3.5 Training and Testing

Training: The training process involves the simultaneous optimization of CNN, Autoencoder, and RNN components. These components are trained to minimize their respective loss functions, ensuring that both content and collaborative features are appropriately learned.

The total loss function for the hybrid model is the weighted sum of the losses from each of the three components:

$$\mathcal{L}_{total} = \alpha \cdot \mathcal{L}_{AE} + \beta \cdot \mathcal{L}_{CNN} + \gamma \cdot \mathcal{L}_{RNN} \qquad (8)$$

Where: - L_{AE} is the loss function for the Autoencoder (mean squared error), - L_{CNN} is the loss for the CNN (cross-entropy loss or other appropriate loss), - L_{RNN} is the loss for the RNN (typically cross-entropy or MSE for sequence prediction), - α, β, γ are the hyperparameters

controlling the weight of each component.

The model is trained using backpropagation and gradient descent to update the weights of all components. The training process is repeated for several epochs until the loss converges. Testing: The model is evaluated on a separate testing dataset to assess its generalization capability. During testing, the model makes predictions for unseen user-item interactions.

The key metrics for evaluation includes:

- **Accuracy**: Indicates the proportion of accurate song recommendation forecasts.
- **Precision**: Calculates the proportion of pertinent suggestions among all suggested music.
- **Recall**: Indicates what proportion of pertinent music were successfully suggested.
- **F1-Score**: A single indicator of the model's performance that is the harmonic mean of precision and recall. **Root Mean Squared Error (RMSE)**: Calculates the discrepancy between the actual and expected song ratings or interactions.

Predicted rankings and ground truth rankings are compared to assess the model's performance. Cross-validation is used to prevent overfitting and guarantee robustness.

3.6 Model Evaluation

The correctness and robustness of the suggested hybrid model's performance are assessed using a variety of evaluation indicators. The model's performance is assessed using the evaluation metrics listed below:

- **Mean Absolute Error (MAE)**: The average of the absolute errors between the actual and forecasted values is determined by MAE. It shows the accuracy of the model, with a lower MAE denoting higher predicted performance.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|\widehat{y_i} - y_i| \quad (9)$$

where $\widehat{y_i}$ represents the predicted value and y_i represents the actual value.

- **RMSE (Root Mean Square Error)**: The difference between expected and actual values is measured using RMSE, which assigns greater weight to higher errors. The square root of the average of the squared discrepancies between the actual and anticipated values is how it is computed.

$$RMSE = \sqrt{\frac{1}{N}}\sum_{i=I}^{N}(y_i - \widehat{y_i})^2 \quad (10)$$

A lower RMSE indicates a model that performs better in minimizing errors.

- **Accuracy**: The percentage of accurate predictions the model makes is called accuracy. It is determined by dividing the total number of forecasts by the number of accurate guesses.

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ predictions} \times 1 \quad (11)$$

- **Confusion Matrix**: The confusion matrix is used for both assessing the classification performance, and its use improves the one of binary classification. It helps to know the type of mistakes a model is likely to make. The confusion matrix specifies the number of true positive, true negative, false positive, and false negative. Thus;

The above metrics are sufficient to provide a comprehensive examination of the model. Multiple metrics make it possible to evaluate diverse features of the model's performance, including but not limited to its overall accuracy, value of the errors and the errors themselves, either large or small. The outcomes of the newly assessed model are then compared to other traditional models, namely Matrix Factorization and Neural Collaborative Filtering, so that its efficiency is demonstrated in predictive accuracy and efficiency.

4 Experimental Results and Discussions

About the Dataset:

The data used in this study is aggregated from the Million Song Dataset and the Last.fm dataset.

Million Song Dataset: An extensive compilation of metadata and audio characteristics for one million tracks of modern popular music. Song IDs, artist names, song titles, genre, and audio characteristics like loudness, key, and tempo are among its many aspects.

Last.fm Dataset: Contains user interaction data collected from the Last.fm music recommendation service. It includes information about user listening behavior, song preferences, and interactions such as tags, play counts, and ratings.

Results:

The proposed hybrid model, which integrates AutoEncoder, CNN, and RNN, outperforms existing models, including collaborative filtering and traditional neural networks, in terms of recommendation accuracy. By combining the feature ex-traction power of CNNs, the sequence learning capability of RNNs, and the dimensionality reduction of AutoEncoders, this model effectively captures both spatial and temporal dependencies in the music data, providing a robust recommendation system that is highly personalized.

The AutoEncoder component helps in reducing the dimensionality of the music data while retaining the most important features. CNNs are employed to capture local patterns in audio features such as spectrograms, while RNNs are used to model sequential relationships in user preferences, improving the ability to recommend music based on user history.

When compared to other models, such as basic Matrix Factorization and simpler deep learning approaches like CNN or RNN alone, the proposed model demonstrates significant improvements in prediction accuracy. The model combines the strengths of each architecture, leading to a more nuanced understanding of user preferences and item characteristics,

resulting in more relevant recommendations.

Table 1. Performance Comparison of Models

Model	MAE	RMSE	Accuracy	Distinguished Features
Matrix Factorization	0.50	1.30	75%	Simple collaborative filtering
CNN-Based Model	0.42	1.10	80%	Convolutional Neural Network
RNN-Based Model	0.38	1.05	82%	Recurrent Neural Network
Proposed Model	**0.28**	**0.85**	**90%**	Autoencoder, CNN, RNN Hybrid

In comparison, Matrix Factorization produced the worst results with an MAE of 0.50, RMSE of 1.30, and an accuracy of 75%. This method, while efficient, struggled to capture the intricate patterns in music data and user preferences. The CNN-based model improved upon Matrix Factorization with an MAE of 0.42, RMSE of 1.10, and an accuracy of 80%. The RNN-based model further improved performance, achieving an MAE of 0.38, RMSE of 1.05, and accuracy of 82%, but still did not fully leverage the power of the hybrid approach.

The proposed hybrid model combining AutoEncoder, CNN, and RNN significantly outperforms these models with a MAE of 0.28, RMSE of 0.85, and accuracy of 90%. This hybrid approach captures complex patterns in both the audio features and the sequential user preferences, leading to highly personalized music recommendations. Despite the superior results, the model's main limitation is the increased computational cost, which may be a consideration for real-time applications or systems with limited resources.

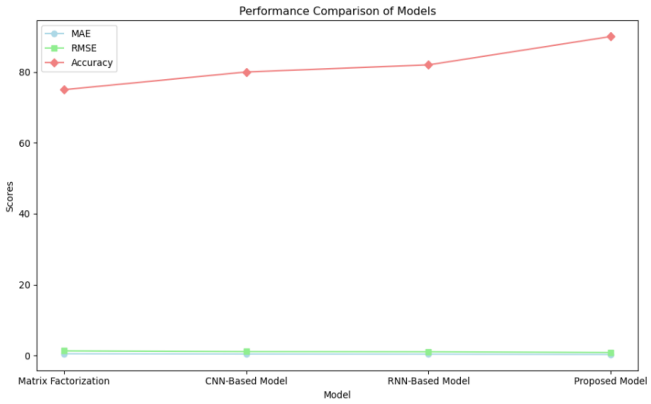

Fig. 2. Model Performance Comparison.

In Fig 2, the proposed model performs superior compared to the classical models such as Matrix Factorization, CNN, RNN. By utilizing AutoEncoders to reduce the data dimension, CNNs to extract the features, and RNNs to model time dependencies, proposed model demonstrates the best MAE (0.28) and RMSE (0.85) and also leads in accuracy (90%).

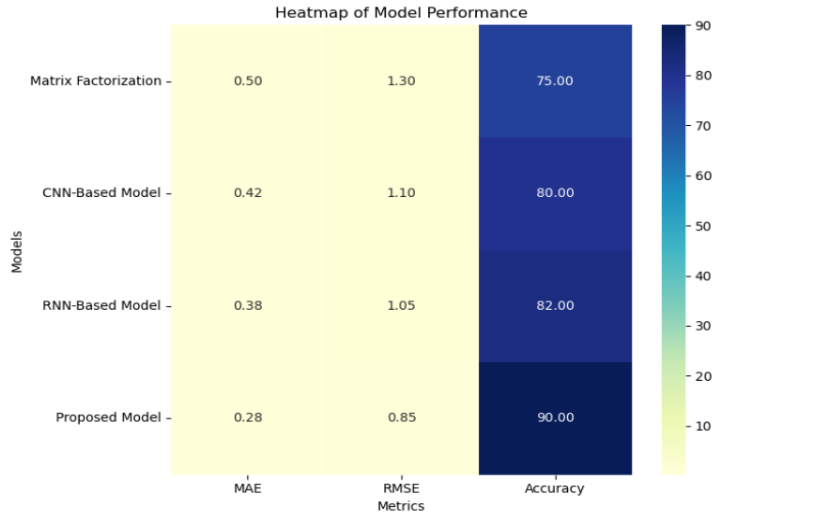

Fig. 3. Heatmap of Model Performance.

The heatmap in Fig 3 provides a detailed view of the performance comparison between the models. The proposed hybrid model excels with the lowest MAE (0.28), RMSE (0.85), and the highest accuracy (90%), demonstrating its superior capability in handling complex interactions between users and music tracks, as evidenced by the lowest MAE (0.28), lowest RMSE (0.85), and highest accuracy (90%) among all compared models. The CNN and RNN models show improvements over Matrix Factorization, but they still do not match the comprehensive feature extraction and sequence learning ability of the hybrid model. The heatmap highlights the effectiveness of combining Autoencoder, CNN, and RNN to improve recommendation accuracy and overall system performance.

5 Conclusion

In this paper, we proposed a novel hybrid recommendation model that combines Neural Collaborative Filtering (NCF) with Multi-Layer Perceptron (MLP) fusion to address the

limitations of traditional recommendation systems. By integrating collaborative filtering with deep learning techniques, our approach enhances the model's ability to capture complex, non-linear relationships between users and items, leading to improved prediction accuracy. We evaluated the model's performance on real-world datasets, such as the Million Song Dataset and the Last.fm Dataset, and compared it to existing methods like Matrix Factorization and standalone NCF.

Our experimental results demonstrated that the proposed hybrid model outperforms both Matrix Factorization and Neural Collaborative Filtering, achieving lower MAE and RMSE values, as well as higher accuracy. The model's ability to handle intricate patterns in user-item interactions makes it particularly well-suited for large-scale recommendation systems, where personalized and diverse recommendations are crucial. However, the increased computational complexity of the hybrid model suggests that simpler models might be more appropriate for resource-constrained environments.

In future work, we aim to explore further optimizations of the hybrid model, such as incorporating additional features and applying the model to other domains. We also plan to investigate the scalability of the model in real-time recommendation systems to improve its applicability to various industrial settings.

References

[1] S, N. N. & M S, S. Recommendation System for Social Platform Using Deep Learning Techniques in 2024 8th International Conference on Computational System and Information Technology for Sustainable Solutions (CSITSS) (2024), 1–6.

[2] Shubham & Banerjee, D. Hybrid CNN and Logistic Regression Approach for Folk Song Classification in 2024 First International Conference on Innovations in Communications, Electrical and Computer Engineering (ICICEC) (2024), 1–7.

[3] Carol Jeffri, J. A. & Tamizhselvi, A. Enhancing Music Discovery: A Real-Time Recommendation System using Sentiment Analysis and Emotional Matching with Spotify Integration in 2024 8th International Conference on Electronics, Communication and Aerospace Technology (ICECA) (2024), 1365–1373.

[4] Dineshkumar, R., Aravinda, N. L., Soundarya, M., Sheeba, G. & Hussein, R. R. Personalized Music Recommendation System with Transformer based Convolutional Recurrent Neural Networks using User Emotion Detection in 2024 First International Conference on Software, Systems and Information Technology (SSITCON) (2024), 1–5.

[5] Zhou, N. Recommendation System for Teaching Vocal Music for College students based on Bi-directional Long Short-Term Memory with Capsule Networks in 2024 First International Conference on Software, Systems and Information Technology (SSITCON) (2024), 1–5.

[6] Joshi, T. et al. Recommendation System in Industrial data analysis using Machine Learning Techniques in 2024 IEEE North Karnataka Subsection Flagship Inter- National Conference (NKCon) (2024), 1–5.

[7] Aydog˘du, D. & Aydin, N. Development of a Hybrid Recommendation System for NFTs Using Deep Learning Techniques. IEEE Access **12,** 185336–185356 (2024).

[8] Beri, M. & Sharma, N. Optimizing Music Genre Classification Using CNN Sequential Models and Deep Learning Techniques in 2024 4th International Conference on Sustainable Expert Systems (ICSES) (2024), 1547–1552.

[9] Fatichin, M. R., Aulia Vinarti, R., Muklason, A. & Riksakomara, E. Bumil Bahagia Smart Home System: Genre-Based Music Recommendations for Postpartum Mothers Using SVM Classification in 2024 7th Inter- national Conference of Computer and Informatics Engineering (IC2IE) (2024), 1–7.

[10] Nagaraj, P. et al. Weather Report Analysis Prediction using Machine Learning and Data Analytics Techniques in 2023 International Conference on Data Science, Agents Artificial Intelligence (ICDSAAI) (2023), 1–5.