# Genre Effect Towards Developing a Multi-modal Movie Recommendation System in Indian Setting

Prabir Mondal, Pulkit Kapoor, Siddharth Singh, Sriparna Saha, *Senior member, IEEE*, Jyoti Prakash Singh, *Senior Member, IEEE,* and Amit Kumar Singh, *Senior member, IEEE*

***Abstract*—The Recommendation System (RS) does not require any introduction nowadays and its role in various platforms as an information filtering agent is very crucial. The personalized movie RS recommends the appropriate movies to the user after analyzing his preferences. Most of the recent studies in this RS basically consider the rating values of the user-movie pairs to evaluate user choice. However, our literature survey found that there is an unexplored area where the user feedback is in ordinal value and the movies are from the Indian regional language based.**

**This paper has presented a multi-modal RS for the Hindi movie dataset where the user preferences over movies are from three different classes and these are, "Like", "Neutral" and "Dislike/ Not watched". The proposed model has employed the utilities of a multi-head cross-attention mechanism and as an input, it has used the extracted audio-video information from the movie trailers. Here the information on Hindi movies and their users has been taken from the Flickscore Dataset. The study evaluates a classification model's effectiveness in two critical aspects: (i) the introduction of a novel metric called "GenreLike-Score (GL-score)," which aligns user genre preferences with movie genres, and (ii) the exploration of various audio-video embeddings. We have done thorough ablation studies using different combinations of these two factors and finally concluded how the GL-score is effective in predicting proper preference for user-movie pair. Besides that, different ways of keyframe extraction techniques as well as their different embedding generation processes have been evaluated in the ablation studies. Our classification model[1] performs well when GL-score is considered and proper audio-video embeddings are determined for the studies.**

***Index Terms*—Flickscore Dataset, GL-score, Hindi Movie Recommendation System, wav2vec, Multi-head cross attention.**

## I. Introduction

The rising volume of internet data and its usage in daily life presents a significant challenge, as users increasingly require quick access to precise information without delay. The task of predicting user preferences has become a pressing concern, particularly within the AI/ML domain. Addressing this issue, Recommendation Systems (RS) have emerged as a leading solution for real-time digital challenges, including music recommendations as highlighted in a study in [1]. Personal viewing devices such as Fire TV, Apple TV, etc., and streaming services in OTT platforms such as SonyLiv, Amazon Prime, Netflix, etc. have necessitated a more focused movie series recommendation to match user preferences.

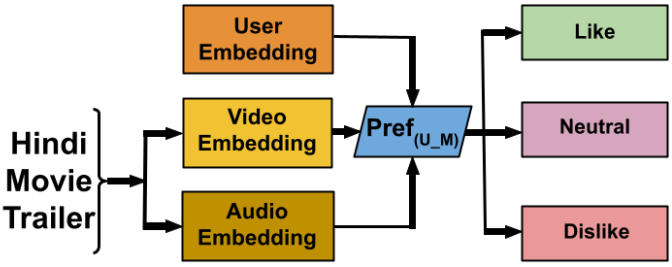


Fig. 1. Graphical Representation of Problem Statement: Given a user-movie pair, the task is to determine any of the three classes ( Like, Dislike, and Neutral )

After surveying the literature on the movie recommendation system, it is noted that most of them are on Hollywood movie-centric datasets and there is scarcity in Indian settings. India is positioned top in making the highest number of movies annually and ranked 3rd in revenue generation in the world. Indian film industry consists of multilingual-multiethnic movies out of which Bollywood, the Hindi language-based movies contribute to bringing the major revenue at the box office. This motivates us to incorporate Bollywood movies in the recommendation system and take the initiative in filling the gap in the research area.

Recommendation system dealing with movies' textual information is common in recent literature, but a movie's audio-visual data is the core information and may help in understanding user preferences also. Analyzing the audio-video content of Hindi movies and understanding the user preference over the genre are the key points of the proposed work. As per the authors in [2], [3], [4], the audio information is a good source of understanding speakers' emotions/ sentiments. Recommending a movie to a user requires three major analyses, *(i) user's genre preference:* What genre does the user like the most and what doesn't, *(ii) Movie's Genre:* What is the genre of the recommended movie, and *(iii) How the Genre suits:* How much the movie's genre will be preferable to the user. To analyze these three key points, we have formulated a GenreLike-Score (GL-score) in our work and reported that it helps the movie recommendation system outstandingly.

In this work, we have attempted to build a classification-

Prabir Mondal, J.P. Singh, and A.K. Singh are with the Department of Computer Sc. and Engineering, National Institute of Technology Patna, India. Pulkit Kapoor is with the Mechanical Department and Sriprana Saha is with the Department of Computer Sc. and Engineering, Indian Institute of Technology Patna, India.
Siddharth Singh is with Electrical Engineering Department, Indian Institute of Technology Jodhpur, India.

Manuscript received April 7, 2023

[1]The code is available: https://github.com/prrabirmondal/Genre-Effect-Towards-Developing-a-Multi-modal-Movie-Recommendation-System-in-Indian-Setting

based movie recommendation system for the Bollywood movie dataset Flickscore [5]. From the movie trailers collected from the online platforms, the audio-video information has been extracted. A thorough investigation has been done to check how the model works in different combinations of audio-video modalities and also to find which embedding process is the best for embedding generation. The proposed method evaluates the GL-score for every user-movie pair to understand how much the user would like the genre of the movie and according to the value, weighted embeddings are passed to our proposed cross-attention-based recommendation system for predicting the preference of the pair.

After an intensive literature survey, we have found recent works are on English movie datasets and textual information mostly. Training and building an RS using the audio-video content of the Indian regional language Hindi movies is the major motive of this study.

On the other hand, the current studies follow the user preference on a 0 to 5 range scale where 0 means *did-not-like* and 5 implies *liked-very-much*. But in real-time scenarios, the user prefers to give feedback in the form of *Dislike/ Like/ Neutral* rather than in numeric form which is time-consuming. Understanding user preference precisely from the ordinal data and recommending movies accurately is really an unmet as well as challenging area of research that we have addressed in our experiment. The proposed model emphasizes the audio-video cross-attention mechanism to incorporate multimedia content in preference prediction. Besides these, the proposed work also tried to conclude that, among the Liked/Disliked/Neutral Movies which combinations are the best for representing a user? Our ablation studies have tried to reach a final decision on this issue.

Above all, an extensive study has been conducted to determine the best audio, video, and user embeddings. It has also been investigated how the genre information of movies and users' genre preferences can help in generating better movie recommendations.

## II. Related works

Existing works on building recommendation systems basically fall under three different types, *(i) Collaborative Filtering, (ii) Content-Based*, and *(iii) Hybrid*. The collaborative approach in recommendation is common in finding user-user, item-item, or user-item similarity. Authors in [6] used the collaborative approach by considering the implicit feedback of other users while recommending movies and the same RS approach has been used in [7] to predict the users' dynamic preferences. A movie Recommendation system by content-based approach [8], [9] has been developed in the work of [10] and it uses the genre content-wise correlations among the MovieLens movies for its model. The content-based movie recommendation system proposed in [11] used movies' textual information. Content-based filtering approach has also been incorporated in [12] in developing a movie recommendation system where actor-based content is used in predicting the user preferences.

Most of the existing works in RS used textual information in predicting user preferences. In [13], authors analyzed the tweets to understand the users' sentiments and the current trends while recommending movies. Authors in [14] predicted the rating values using a deep neural network-based trusted filter (DNN-filter) and utilized the analyzing potential of deep learning for multimodal data. A deep learning-based model for movie recommendation has also been introduced by authors in [15] where textual information is used in the analysis. Multimedia content-based information is effective in the movie recommendation system proposed by authors in [16] and in [17] authors considered user feedback rather than a user profile and fused the multi-modal information by incorporating the attention mechanism.

Be its ARTIST model in [18], the user preferences are predicted after tracking their emotions extracted from their reactions which were captured by different sensors. Similarly, the regression-based model was developed by authors in [19] where a multi-modal extended MovieLens dataset was used.

Extensive works are also being done in the RS area with the incorporation of Graph Neural Network [20] based approach. By introducing the Graph Attention Network [21] and adversarial network in the proposed model, authors in [22], [23] tried to handle the cold-start problem [24] over the multi-modal MovieLens dataset. Similarly in [25], the graph attention is used in uniforming the user's as well as the item's auxiliary information for its RS. Authors in [26] used the Graph convolutional Network [27] for better movie representation in their proposed RS.

After going through the literature, it is found that most of the works are regression-based on Hollywood movie datasets. This motivates us to investigate how RS performs in classification in Indian Settings.

**Research Gap:** Existing studies in RS have mainly focused on using movie's textual information as input, neglecting easily available audio-video data. To address this gap, we have incorporated audio-video content independently, excluding the textual modality. Additionally, while the traditional approach revolves around rating prediction, our research reveals a lack of classification models. We have also noticed that, typically, the movie embeddings are blindly fed as input to the model for different users, disregarding their distinct preferences for common movies. But, user-specific item embeddings are necessary. Furthermore, users generally consider genres when selecting movies but, there are fewer works [28] in genre-based recommendation systems. Along with these, no previous work has focused on RS development in the context of Indian settings. Our proposed approach tackles these issues and introduces a novel method for generating user embeddings, multimodality fusion, and developing an Indian regional-language-based Hindi-Movie-RS.

## III. Problem Statement

*Like (L), Dislike (D)*, and *Neutral (N)*, among these three classes our classification-based model predicts one that implies the preference of the user for the candidate movie. The pictorial representation of the problem statement has been shown in figure 1.

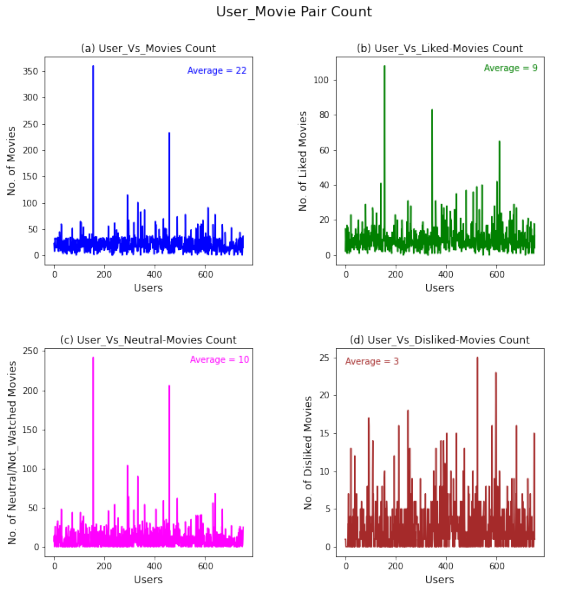


Fig. 2. Feedback-wise User-Movie count: (a) *User-vs-Movie Count for all feedback*, (b) *User-vs-Movie Count for the Liked Movies*, (c) *User-vs-Movie Count for the Neutral/Not-Watched Movies*, (d) *User-vs-Movie Count for the Dislike Movies*

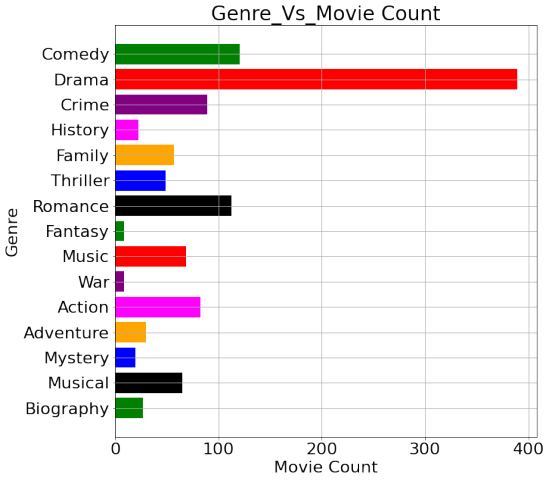


Fig. 3. Genre Distribution: Genre-wise Hindi movie count for Flickscore Dataset.

The proposed model takes the user-movie pair as input and in the output part, it notifies one of the classes among the three as the user's preference for the candidate input movie.

$$F(fuse(M_v, M_a, U_{emb})) = Pref_{(U\_M)} \quad (1)$$

Where $Pref_{(U\_M)} \in$ (*Dislike, Like, Neutral*). Our methodology entails the fusion of both video ($M_v$) and audio ($M_a$) modalities for representing a movie ($M$). This fused representation is then combined with the user's embedding ($U_{emb}$) in our proposed model to predict the user's preference ($Pref_{(U_M)}$) for this particular movie-user pairing. During experimentation, we combine the two movie modalities ($M_v$ and $M_a$) and employ a learned function denoted as $F$ to predict the user's preference for the movie, as shown in Equation 1. Beyond model construction, this research also encompasses the evaluation of the model's performance across diverse scenarios and the identification of specific corner cases. These efforts collectively contribute to the refinement of a recommendation system tailored for Hindi movies.

## IV. Dataset

Existing movie datasets are mostly based on Hollywood movies and user preferences are from a specific rating range. However, we followed our goal in this research and used the Flickscore [5] dataset. This dataset includes meta-information about 2,851 movies of 15 genres *(Biography, Musical, Mystery, Adventure, Action, War, Music, Fantasy, Romance, Thriller, Family, History, Crime, Drama, and Comedy)* The movies included in this dataset are of 18 different Indian regional languages *(Hindi, Bengali, Assamese, Tamil, Nepali, Punjabi, Rajasthani, Malayalam, Bhojpuri, Kannada, Haryanvi, Manipuri, Urdu, Marathi, Telugu, Oriya, Gujarati, Konkani)*. It also includes meta information for 919 viewers. The Hindi movies and their users are more in counting than other language-based movies of the Flickscore dataset. Besides that, the trailers of most Hindi movies are available online. For the availability of Hindi movie information, we have considered these movies only from the dataset for our proposed experiment and collected their trailers to obtain their audio-video content. Users in this dataset scored movies using one of three values: 0 for *neutral or not viewed*, -1 for *dislike*, and 1 for *like*. This dataset does not include traditional rating scales like 1 to 5 stars. **Video-Audio Information:** A full-length Hindi film typically lasts between 120 and 180 minutes. At a frame rate of 24 fps, there are at least $120 \times 60 \times 24 = 172,800$ frames in a full-length Hindi movie.

According to the aforementioned estimate, processing the visual and audio information of a full-length movie is highly time-consuming and storage-intensive. Therefore, inspired by [29], we used movie trailers instead of full-length movies. Movie trailers convey the overall theme of a movie in a very short period of time, using about 2% of the frames and audio of the full-length movie.

Although this dataset contains 615 Hindi movies, some of the Hindi movies are too old to find trailers on popular online streaming platforms like *YouTube:* (https://www.youtube.com/), *IMDb:* (https://www.imdb.com/), *RottenTomatoes:* (https://www.rottentomatoes.com/), etc. Therefore, our dataset contains 510 Hindi movies with audio and video information extracted from their trailers.

There are 753 users and a total of 16,667 user-movie pairs with feedback for the 510 Hindi movies. The user-movie pair is a triplet that includes the *user-id, the movie-id, and the user's input*. There are 1887 user-movie pairs for liked movies (*L*), 7813 for neutral or not watched movies (*N*), and 6967 for disliked movies (*D*) in the Hindi Movies multi-modal dataset. According to Figure 2a., each user has given feedback on an average of 22 movies, with 9 for Liked-Movies, 10 for Neutral/Not-Watched Movies, and 3 for Disliked Movies, as indicated in Figure 2b., Figure 2c. and Figure 2d., respectively. Figure 3 shows the number of Hindi movies in each genre.

## V. METHODOLOGY

The proposed model predicts the preference of a user for a movie by taking the embeddings of the user and the movie as input. But generating the input embedding and finally predicting the preference class for the input pair requires a couple of steps as stated below.

### A. *Video Embedding*

For generating video embedding for our experiments we have used our video embedder shown in Figure 4(a). The key frames from the movie trailer are extracted first and then passed to the video embedder for generating the embedding. In the following sections, detailed explanations for generating the video embedding have been presented.

*1) **Frame Extraction**:* We have gone through three different techniques for extracting the 16 key frames from the trailer for the input of TimeSformer[30] and finally chose the best one among them by analyzing the result of the proposed model. Here are the details of the techniques:

*a) **Video Frames in Equal Interval**:* Here, we have extracted the 16 frames from each movie trailer following the systematic sampling. 16 frames are selected uniformly, meaning the frames are extracted at equal intervals throughout the trailer. As it gives the best frames for representing a video, we have considered this technique for video frame extraction in the proposed model.

*b) **Video Frames from Each Scene**:* In this method, each scene from every trailer is detected first using the Python open source library PySceneDetect[2] and then a single frame from each scene is extracted using the save_images function of the PySceneDetect.

Since different movies have a different number of scenes, the number of extracted frames will also be different. So, from the extracted frames, 16 frames that were in the middle position of the trailer were collected as keyframes. In this way, we try to neglect the frames which lie either at the beginning or the end of the movie trailer, since in most cases they are blank frames or contain text information only.

*c) **Video Frames by applying YOLOv5**:* Here also the scenes are detected first from the trailer using the PySceneDetect module of Python and then the object detection algorithm YOLOv5 [31], [32] has been applied to all the frames of each scene. Finally, the top 16 frames having the maximum number of objects in each scene are collected as keyframes.

*2) **Video Embedder**:* The Video Embedder has been used for genre classification of the movie and extracting the genre-based task-specific video embedding. In our experiment, we have used *TimeSformer* [30] for extracting the genre-based video embedding of Flickscore movie trailers. The keyframes extracted using the above-described methods are passed to the TimeSformer model for getting the genre-based task-specific video embedding. We used transfer learning to adapt the TimeSformer model, which was pre-trained on the K-600 dataset for video genre classification, to our task. We made many adjustments to the model to build the embedder,

[2]https://scenedetect.com/en/latest/

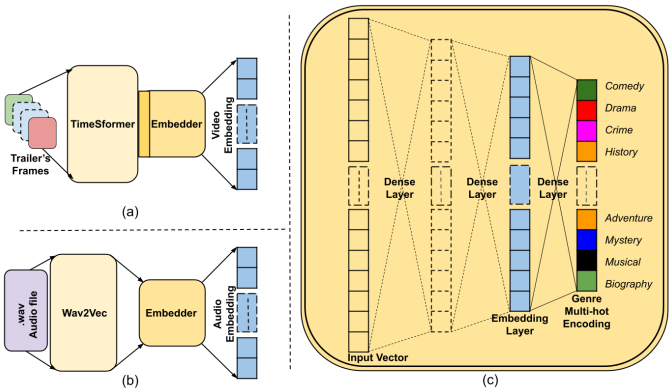


Fig. 4. Embedding technique: (a) *Video Embedding process*, (b) *Audio Embedding process*, (c) *Schematic architecture of the Embedder.*

as shown in Figure 4(c). First, we deleted the TimeSformer model's classification layers and inserted a new layer with 512 neurons as the embedding layer of our video embedder. Next, a classification layer with 15 neurons was added to do the multi-label classification. This was important since categorizing videos into different genres is frequently a multi-label problem, meaning that a video may fit into more than one genre.

The fine-tuning process includes training the newly added layers of the modified pre-trained TimeSformer model with our trailers' keyframes. After training, the video embedding of the movie trailer is finally recovered from the 512-dimensional embedding layer.

We have collected the video embeddings of the frames extracted by each extraction technique and finally considered the technique that gives the best preference prediction in the main model.

### B. *Audio Embedding*

The audio files of 510 Hindi movie trailers were first extracted in *.wav* format and then transformed to vector form using *wav2vec2* [33]. We utilized wav2vec2 as a feature extractor, which produces a 512-dimensional feature vector.

For generating Genre-based task-specific audio embedding, we have used the dense layer embedder as shown in Figure 4(c). The embedder has 3 dense layers with neurons of dimensions 512, 512, and 512, respectively. The activation function used is (tanh) in all three dense layers. The last layer is the classification layer with 15 neurons and with Sigmoid as an activation function. The input to the model is the feature vector of dimension 512 that we get through *wav2vec2* and for predicting 15 genres, the output is fifteen-dimensional sigmoid output. Finally, the vector of 512 dimensions from the *Embedding Layer* (second last layer) of the embedder is extracted as the audio embedding of the movie trailer.

### C. *Embedder*

Figure 4(c) shows the Embedder which is a sequence of neurons' dense layers and has been used in audio and video embedding. In its last layer, the 15-dimensional genre multi-hot encoding for a movie is predicted from the audio or video

input information. The number of layers is a hyperparameter and the vector from the second last layer of it is taken out as the corresponding modality embedding of the movie.

### D. GL-score

Three steps are followed in GenreLike-Score (GL-score) calculation for a user-movie(U_M) pair,

*1) **User's Genre Embedding Generation**:* In this step, three sets, i.e., $L_g, N_g, D_g$ are formed where set $L_g$ consists of the genre multi-hot encoding of all the movies **liked** by the user $U$. Similarly, set $N_g$ and set $D_g$ are the collection of genre encoding of all the movies rated as **Neutral** and **Dislike**, respectively by the user U. Finally as described in Figure 5, by following Equation 2 the user's genre encoding for like ($U^L$), neutral ($U^N$) and dislike ($U^D$) are generated.

$$\mathbf{U}^K_{j=(1-to-15)} = \sum_{i=1}^{n} \mathbf{K}_{ij} \tag{2}$$

Where $\mathbf{K} \in (L_g, N_g, D_g)$ and $\mathbf{n}$ is the total number of movies present in K.

*2) **like, neutral, and dislike score calculation**:* After generating the vectors $U^L$, $U^N$ and $U^D$ mentioned in Equation 2, the dot products between the vector and the multi-hot genre encoding ($M_g$) of the movie ($M$) are calculated as shown in Equations 3, 4, 5. Finally the like, neutral, and dislike scores of the user-movie pair are obtained and are termed as l-score, n-core, and d-score, respectively.

$$\text{l-score} = (M_g).(U^L)^T \tag{3}$$

$$\text{n-score} = (M_g).(U^N)^T \tag{4}$$

$$\text{d-score} = (M_g).(U^D)^T \tag{5}$$

*3) **Computing GL-score**:* The final GL-score for the user-movie pair is calculated from the corresponding like, neutral and dislike score by following Equation 6. The GL-score quantifies, how much the user likes the recommended movie's genre. All the parameters used in calculating the GL-score have been described in Table I.

$$\text{GL-score}_{(U\_M)} = \frac{\text{l-score} + (\text{n-score}/2)}{\text{l-score} + \text{n-score} + \text{d-score}} \tag{6}$$

### E. User Embedding

For generating the user embedding, the movies($M_i^U$) rated by the user (U) and the GL-scores(GL-Score$_{U_M_i}$) for those user-movie pairs are considered. All the movie embeddings are multiplied with corresponding GL-scores and finally, the average of these new embeddings is used as the user embedding($U_{emb}$). Equation 7 shows the user embedding generation formulation.

$$U_{emb} = \frac{1}{n} \sum_{i=1}^{n} \text{GL-score}_{(U\_M_i)} \times (M_i^U) \tag{7}$$

Here, $U_{emb}$ is the user embedding, GL-score$_{(U_M_i)}$ is the GL-score for the $(U\_M_i)$ pair, $M_i^U$ is the $i^{th}$ movie $(M_i)$ rated by the user $(U)$, and $n$ is the total number of movies rated by the user $U$. By following the conclusion of our thorough

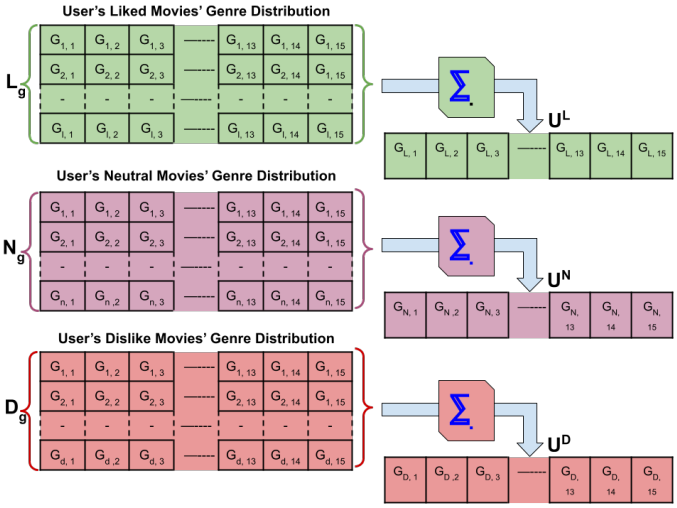


Fig. 5. User's Genre Encoding from rated movies: ($U^L$)**:** *User's genre encoding from Liked movies*, ($U^N$)**:** *User's genre encoding from Neutral movies*, ($U^D$)**:** *User's genre encoding from Disliked movies*

experimentations, for better user embedding generation, we have considered movies that are liked by the user. So, $n$ is the total number of Liked movies by the user. For ablation studies, different combinations of user-rated movies are taken and their results are reported in the *Result and Analysis section*. The different combinations are, *(i) L movies, (ii) (L+D) movies, (iii) (L+N) movies, and (iv) (L+N+D) movies*.

### F. Main Model

To understand the user-movie pair preferences, cross attention between the user and the movie is highly desired. Here, the goal is to make a prediction of whether the user would give the movie a like, dislike, or neutral rating. To achieve this, we employ cross-attention.

In the proposed main model, the attention mechanism presented in [34] has been employed to consider the attention factor in the user-movie pair while predicting the pair preference. The (Q, K, V) triplet for both user and movie in the model participated crosswise and generated the corresponding context vectors using the mechanism formulated in Equation 8 and Equation 9.

In the attention part, the model has used two different attention mechanisms, *(i) cross-attention* and *(ii) self-attention (in Fusion block)*. Figure 6 presents the main model architecture where initially the cross attention between user-movie is performed and then the generated attention based two vectors $\hat{\mathbf{U}}$ and $\hat{\mathbf{M}}$ are concatenated to pass to the self-attention layer of fusion block. Finally, the output from the self-attention is passed through a feed-forward and dense layer to predict the preference. In the model, the attention blocks have used a four-head attention mechanism.

$$Attention\_weight = Softmax\left(\frac{Q.K^T}{\sqrt{d_k}}\right) \tag{8}$$

$$Context\_vector = Attention\_weight * V \tag{9}$$

where $d_k$ is the dimension of the vectors K.

TABLE I
DESCRIPTION OF PARAMETERS USED IN GL-SCORE CALCULATIONS

| Parameters | Description | Parameters | Description | Parameters | Description |
|---|---|---|---|---|---|
| $L_g$ | Set of genre embeddings of movies liked by a user | $U^D$ | Element-wise sum of vectors in $D_g$ for user U | $(U)^T$ | Transpose of vector U |
| $N_g$ | Set of genre embeddings of movies to which user is neutral | M | Movie | GL-score$_{(U_M)}$ | GL-score between user (U) and Movie (M) |
| $D_g$ | Set of genre embeddings of movies disliked by a user | $M_g$ | Genre embedding of movie (M) | GL-score$_{(U_M_i)}$ | GL-score between user (U) and $i^{th}$ Movie ($M_i$) |
| U | User | l-score | Dot product of vectors $M_g$ and $U^L$ | $M_i^U$ | $i^{th}$ movie ($M_i$) watched by user (U) |
| $U^L$ | The element-wise sum of vectors in $L_g$ for user U | n-score | Dot product of vectors $M_g$ and $U^N$ | $U_{emb}$ | User embedding |
| $U^N$ | The element-wise sum of vectors in $N_g$ for user U | d-score | Dot product of vectors $M_g$ and $U^D$ | $M'$ | Movie embedding after applying GL-score |

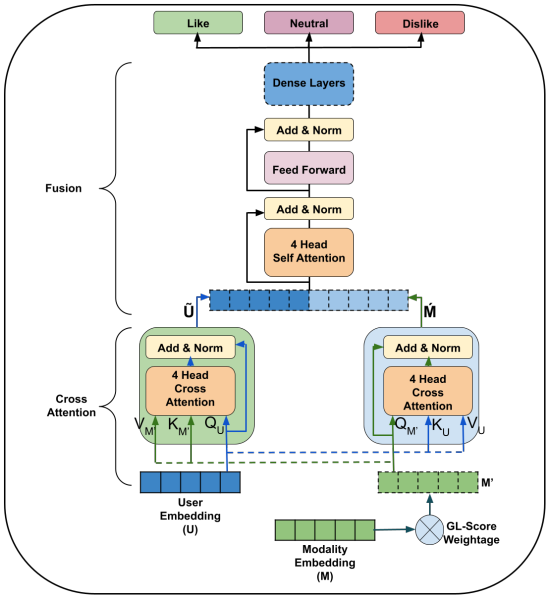


Fig. 6. Cross Attention-based proposed main model of Movie Recommendation.

*1) Model's Inputs and Output:* The main model takes two vectors as input, *(i) user embedding ($U_{emb}$)* and *(ii) modality embedding($\boldsymbol{M'}$)*. The modality embedding is either audio or video embedding of the movie but it is not passed directly to the model. The GL-score is applied to modality embedding as shown in Equation 10 and the weighted modality embedding is passed as input to the model.

$$\mathbf{M'} = \mathbf{M} \times \text{GL-score}_{(U\_M)} \tag{10}$$

Here, $\mathbf{M}$ is the modality embedding and GL-Score$_{(U_M)}$ is the GL-Score between user $U$ and movie $M$ pair.

## VI. RESULTS AND DISCUSSION

This section describes the experimental setup and the detailed analysis of the experimental results.

### A. *Experimental Setup*

*1) **Audio Video Embedding**:* The audio-video embedder is trained in 100 epochs with Adam optimizer, categorical cross-entropy loss [35] function with accuracy as the performance metric. Except for the last layer of the embedder, all layers use the tanh activation function whereas Sigmoid is used in the output layer.

*2) **Main Model Training Setup**:* The results in the main model are generated by the stratified five-fold cross-validation [36]. With the Softmax activation function in the last layer, Adam optimizer, and categorical cross-entropy loss function, the model is trained in 150 epochs with patience value = 5. In the main model, there are feed-forward and dense layers in the fusion block. The layer dimensions in feed forward: $[d_{Input}] \rightarrow [\frac{d_{Input}}{2}] \xrightarrow{ReLU} [d_{Input}]$ and that in Dense layer: $[d_{Input}] \rightarrow [\frac{d_{Input}}{2}] \xrightarrow{ReLU} Softmax(d_{Output})$. Here, $[d_{Input} = 512]$ and $[d_{Output} = 3]$ are the respective dimensions of the Input and Output of the model.

*3) **Train Test Split**:* For audio-video embedding generation, the dataset of 510 Hindi movies is split into 450:60 as a training-testing split. The main model is trained in a five-fold stratified cross-validation setting but before that, the user-movie-preference triplet for the *Dislike (D)* class has been up-sampled to 6,625 from 1,887 to make the dataset balanced. Here the split of training-testing is 80:20 whereas the training: validation split is 90:10.

*4) **Performance Metrices**:* Performances of the classification-based main model are represented in Tables II, III, and IV in the percentage of Precision (**P**), Recall (**R**), F1-score (**F1**), Accuracy (**Acc.**), and their weighted average accuracy (**Wt. Avg. Acc.**) All these tables have been divided into two parts, *(i) With GL-score:* when the GL-score is employed in the preference prediction and *(ii) Without GL-score:* when the GL-score is not considered in both user and modality embedding. Here the generated embedding is directly fed to the model in *Without GL-score* approach. In these tables, the model's performance on different types of user embeddings has been shown with the terminologies, ***L:** Like*, ***L+D:** Like+Dislike*, ***L+N:** Like+Neutral* and ***L+D+N:** Like+Dislike+Neutral*. *Like* implies all the movies liked by the user. Similarly *Dislike and Neutral* are the respective disliked movies and neutral/not watched movies of the user. The results in these tables are the average of five-fold with the standard deviation of the best model trained in 150 epochs. The highest values in the comparisons are made bold in the tables and the comparison is based on the weighted average accuracy and F1-score. The higher the weighted average

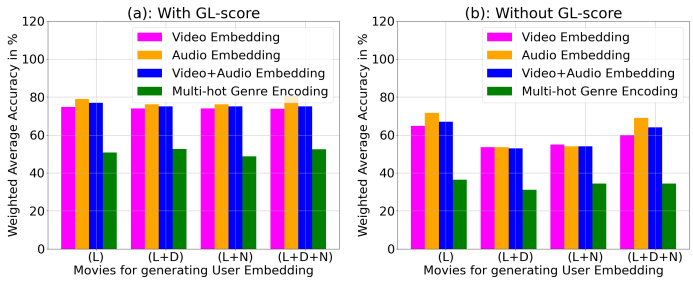


Fig. 7. Weighted average accuracy from different Embedding ($Video, Audio, Audio + Video, Genres' multi\_hotEncoding$): (a) *Results with GL-score*, (b) *Results without GL-score*

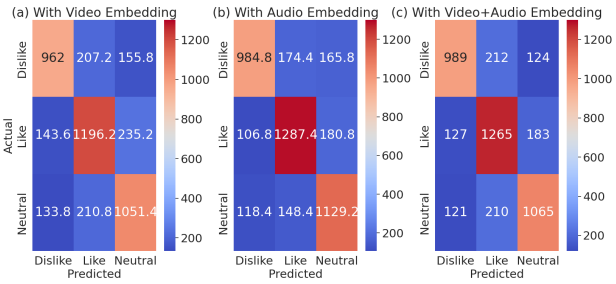


Fig. 8. Confusion Matrix: (a) *Video Embedding*, (b) *Audio Embedding*, (c) *Video+Audio Embedding*

accuracy the better the model's performance is. For the other models (audio, and video embedder), the accuracy metric has been considered for measuring the model's performance. All the results proposed in this study are statistically significant.

### B. *Result Analysis*

The cross-attention-based classification model takes GL-score weighted embedding pair (user and modality embedding of movie's trailer) and tries to predict the preference of the pair. We have tested the model in single modality and bi-modality. In a single modality, the weighted video or audio embedding is passed individually to the model with its corresponding user embedding. In bi-modality, the (video+user) and (audio+user) embeddings are passed as input of the model.

*1) **Video Modality Performance**:* We have used different techniques for extracting the keyframes from trailers as per our requirement and finally, it has been observed that the embeddings of frames extracted in equal intervals are best in representing the video modality of the movie's trailer. Table II shows the model's performance when the video embeddings are generated from the frames extracted by the method presented in section V-A1a. Here the GL-score and without GL-score wise model's performances have been tabulated with different types of user embeddings. It shows that the GL-score has a positive impact on the result and the model performs well when the user embedding is formed by only considering the **Liked (L)** movies of the user. When GL-score and **L** movies are considered together the weighted average accuracy is 10% higher than that of the without GL-score approach.

*2) **Audio Modality Performance**:* The performance of the audio embedding generated by our audio embedder has been reported in Table III. Here also the results show how the model performs when different types of user embeddings and GL-score are considered. It is observed that this score supports the model strongly. The value of performance parameters for Like user embedding is lower than that of (L+D+N) user embedding but not in a very high margin and the weighted average accuracy for L+GL-score is 7.4% higher than that of without GL-score approach.

*3) **Multi-Modality Performance**:* For testing the performance of bi-modality or multi-modality i.e.,(Video+Audio) over the proposed model we have taken two pairs of input *(i) User-Video* and *(ii) User-Audio*. These two pairs of input are passed individually to the model and the outputs from the cross-attention section are concatenated and passed to the Fusion block for generating the preference. Figure 7 shows the model's performance over bi-modality and uni-modality. It is observed that the audio modality (the orange color bar) outperforms all the modalities and the bi-modality performs better than the video modality. Here also the GL-score has a positive effect on bi-modality. Figure 8 illustrates a heatmap of the confusion matrix across various modalities and it has been observed that the model predicts the liked movies better than those of the disliked and neutral/not watched movies. Predicting liked movies is one of the desiring concerns for recommendation systems and hence the model has tried to meet the requirements successfully.

*4) **GL-score Effect**:* Our all experiments have proved enough that the cumulative GenreLike-score (GL-Score) has a strong positive impact in estimating the user-movie pair's preference. It is very prominent from Table II, III, IV, and Figure 7 that all the performance metric values (weighted average accuracy, F1-score) for GL-score based approach are higher than the without GL-score based method. So, it is strongly proved that the genre information of movies and genre preference of users play a strong role in better recommendation.

*5) **Best User embedding**:* The four different combinations of movies i.e., *[L, (L+D), (L+N), and (L+D+N)]* are taken for generating different types of user embeddings. From Tables II, III, and IV, it is concluded that in major cases the user embedding is the best when it is generated by considering only the liked movies of the user. In Table III and IV(B) the results for user embedding generated using the liked movies are not the highest but close to the highest values.

### C. *Ablation Studies*

To conclude the best approaches in the experiments, a number of ablation studies have been conducted and those are reported as follows,

*1) **Video Modalities**:* We have three different approaches for extracting keyframes from movie trailers. Table II shows the model's outputs when the considered frames are generated by the method described in section V-A1a. Table IV(A) and Table IV(B) present the model's outputs when frames are extracted by the methods described in section V-A1b and V-A1c, respectively.

From the above-mentioned tables, it is observed that the performance metric values (weighted average accuracy, F1-score) in Table II and in the section IV(A) are quite close when GL-score is applied and these values are higher than that of Table IV(B).

For considering the best and highest performance we have chosen the frame extraction approach presented in section V-A1a.

Table V shows the model's performance when it considers the video embeddings of frames extracted using equal interval techniques and embeddings are taken from the second last layer (Average Pooling layer) of ResNet-50, the base model. Comparing the results with Table II, it justifies the application of TimeSfromer in video embedding generation.

*2) **Audio Modalities**:* Table III(B) presents the model's performance when audio's wav2vec2 embedding is directly passed to the model. Here, for audio embedding, the embedder shown in Figure 4 is not used and the result is compared with wav2vec2+genre based audio embedding. Table III shows the comparative results of these two embeddings and finally, it is apparent that our embedder helps in generating better audio representation. When the GL-score is taken into account, the result for the wav2vec2+genre based audio embedding is 4.4% higher than that of only wav2vec2 audio embedding.

*3) **Audio-Video Embedding vs. Genre Encoding**:* In the Flickscore dataset, every movie has 15-dimensional genre information that has been used in our embedder presented in Figure 4c. In the ablation study, we also tried to find out the importance of audio-video embedding. Similar to audio-video embedding, we passed the movie's 15-dimensional multi-hot genre encoding as input with its corresponding user embedding to the model and checked the necessity of audio-video embedding. The green bar in Figure 7 represents the weighted average result when the movie's encoding is the multi-hot genre encoding and it is concluded that the audio-video embeddings outperform the genre multi-hot encoding in every scenario.

*4) **How does ANN-based Classification model work?**:* In the ablation study, we have compared our model performance with the Artificial Neural Network (ANN) based classification model whose layer-wise architecture is as follows,

$\left[\begin{array}{c} 512 \rightarrow 256 \xrightarrow{tanh} 128 \xrightarrow{tanh} Tanh(64) \\ 512 \rightarrow 256 \xrightarrow{tanh} 128 \xrightarrow{tanh} tanh(64) \end{array}\right] \Rightarrow$ *Classification_Block*, and *Classification_Block:*

$128 \xrightarrow{tanh} 64 \xrightarrow{tanh} 32 \xrightarrow{tanh} Softmax(3)$

Here, the input of the ANN model is user and movie embeddings of dimension 512. As the model predicts the like, neutral, and dislike classes, it has 3 neurons in the output layer. Table VI shows different user embedding-wise Proposed Model(PM) and ANN models' performances over different modalities of movies. In this experiment, the video embedding of frames extracted by the technique described in section V-A1a has been used. In audio modality, our proposed wav2vec2+genre based audio embedding has been chosen. It shows that in every scenario the performance of PM is higher than that of ANN with respect to Wt. Avg. accuracy.

### D. Error Analysis

In addition to the favorable effect of GL-score, the above analysis shows that the combined modality does not work well in the proposed model. This could be the for many factors that must be addressed. The reasons could be, *(i) Poor Video quality and Video Frame Extraction Technique:* The movies in the dataset are quite old and thus the video quality of trailers is also very poor. Removing the shadow effect from old movies' frames as presented in [37] or adding new movies to the dataset might be helpful in standardizing the video dataset. Furthermore, keyframes that indicate the main idea of the film may also be lost since a specific number of frames are taken from the movie trailer at regular intervals. *(ii) Modality Fusion Technique:* The concatenation fusion approach was taken into account here. This method is fairly straightforward. Some new fusion techniques can be included in future studies. *(iii) Low Volume Dataset:* There are only 510 videos containing multi-modal information and 753 users in our dataset. The model is not being sufficiently assisted by this small dataset in determining the correlations between user-movie pair and user preference.

### E. Major Results and Limitations

The experiments conducted in this study show the following findings: (i) Video embeddings of the frames generated through a systematic sampling, yield superior results compared to YOLOv5-based frames as well as frames from every scene. (ii) Our proposed GL-score positively influences both unimodality and bimodality. (iii) Ablation studies reveal that Liked-movies, rather than Disliked/Neutral movies, are optimal for generating user embeddings. (iv) Among all modalities, Audio outperforms both Bimodality (Audio+Video) and Video. Bimodality performs better than Video alone. (v) Converting wav2vec2 embeddings to genre-based audio embeddings using an audio embedder improves results for the Audio modality. (vi) Using audio-video embeddings instead of genre multi-hot encoding is justified.

Limitations include the smaller size of the dataset and bimodality underperforming unimodality (audio). Addressing these limitations is crucial for future improvements.

## VII. Conclusion

In this paper, we have developed a classification-based Hindi movie recommendation System and tested it over the Flickscore Dataset. This model employed the cross attention between user-movie information and considered GL-score to evaluate users' genre preferences. Along with the equal interval key frames' embeddings and wav2vec2+genre-based audio embedding, the cumulative GL-score helps the model outstandingly in preference prediction.

The bi-modality does not outperform the uni-modality and hence more effort is required to make the model multi-modal. Besides all, incorporating the textual information and experimenting with the model performance over large and different datasets is one of our future targets. This investigation on the new model and the new dataset does not have any competitive work in state-of-the-art. The conclusion of the investigation

TABLE II
VIDEO EMBEDDING RESULT: DIFFERENT USER EMBEDDING WISE GL-SCORE IMPACT ON THE RESULTS WHEN FRAMES ARE EXTRACTED IN EQUAL INTERVAL

| *Movies used in generating USER's Embedding* | **With GL-score** | | | | | **Without GL-score** | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | **P** | **R** | **F1 score** | **Acc.** | **Wt. Avg. Acc.** | **P** | **R** | **F1 score** | **Acc.** | **Wt. Avg. Acc.** |
| **L** | **74.711** ±0.33 | **74.711** ±0.33 | **74.711** ±0.33 | **74.711** ±0.33 | **74.80** ±0.40 | **64.772** ±0.59 | **64.772** ±0.59 | **64.772** ±0.59 | **64.772** ±0.59 | **64.800** ±0.48 |
| **L+D** | 74.278 ±0.15 | 74.278 ±0.15 | 74.278 ±0.15 | 74.278 ±0.15 | 74.000 ±0.00 | 54.939 ±0.29 | 54.939 ±0.29 | 54.939 ±0.29 | 54.939 ±0.29 | 55.000 ±0.75 |
| **L+N** | 74.083 ±0.22 | 74.083 ±0.22 | 74.083 ±0.22 | 74.083 ±0.22 | 74.000 ±0.00 | 55.028 ±0.23 | 55.028 ±0.23 | 55.028 ±0.23 | 55.028 ±0.23 | 55.000 ±0.40 |
| **L+D+N** | 73.883 ±0.38 | 73.883 ±0.38 | 73.883 ±0.38 | 73.883 ±0.38 | 73.800 ±0.40 | 59.898 ±0.87 | 59.898 ±0.87 | 59.898 ±0.87 | 59.898 ±0.87 | 59.800 ±0.98 |

TABLE III
AUDIO EMBEDDING RESULT: DIFFERENT USER-EMBEDDING WISE GL-SCORE IMPACT ON THE RESULTS WHEN *(A)* WAV2VEC AND GENRE PREDICTION METHODS ARE APPLIED AND *(B)* ONLY WAV2VEC METHOD APPLIED FOR GENERATING AUDIO EMBEDDINGS

| *Movies used in generating USER's Embedding* | **With GL-score** | | | | | **Without GL-score** | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | **P** | **R** | **F1 score** | **Acc.** | **Wt. Avg. Acc.** | **P** | **R** | **F1 score** | **Acc.** | **Wt. Avg. Acc.** |
| ***(A) Audio Embedding: [wav2vec + Genre Prediction]*** | | | | | | | | | | |
| **L** | 79.176 ±0.61 | 79.176 ±0.61 | 79.176 ±0.61 | 79.176 ±0.61 | 79.000 ±0.63 | **71.443** ±0.36 | **71.443** ±0.36 | **71.443** ±0.36 | **71.443** ±0.36 | **71.600** ±0.49 |
| **L+D** | 76.215 ±0.43 | 76.215 ±0.43 | 76.215 ±0.43 | 76.215 ±0.43 | 76.200 ±0.40 | 54.637 ±0.58 | 54.637 ±0.58 | 54.637 ±0.58 | 54.637 ±0.58 | 53.600 ±0.49 |
| **L+N** | 76.271 ±0.23 | 76.271 ±0.23 | 76.271 ±0.23 | 76.271 ±0.23 | 76.200 ±0.40 | 55.056 ±0.27 | 55.056 ±0.27 | 55.056 ±0.27 | 55.056 ±0.27 | 54.000 ±0.00 |
| **L+D+N** | **80.559** ±0.17 | **80.559** ±0.17 | **80.559** ±0.17 | **80.559** ±0.17 | **80.600** ±0.49 | 69.046 ±0.50 | 69.046 ±0.50 | 69.046 ±0.50 | 69.046 ±0.50 | 69.000 ±0.63 |
| ***(B) Audio Embedding: [wav2vec only]*** | | | | | | | | | | |
| **L** | 74.558 ±0.16 | 74.558 ±0.16 | 74.558 ±0.16 | 74.558 ±0.16 | 74.600 ±0.49 | **58.259** ±0.31 | **58.259** ±0.31 | **58.259** ±0.31 | **58.259** ±0.31 | **57.800** ±0.40 |
| **L+D** | 74.697 ±0.12 | 74.697 ±0.12 | 74.697 ±0.12 | 74.697 ±0.12 | **75.000** ±0.00 | 54.614 ±0.38 | 54.614 ±0.38 | 54.614 ±0.38 | 54.614 ±0.38 | 53.400 ±0.49 |
| **L+N** | **74.735** ±0.16 | **74.735** ±0.16 | **74.735** ±0.16 | **74.735** ±0.16 | 74.800 ±0.40 | 54.539 ±0.23 | 54.539 ±0.23 | 54.539 ±0.23 | 54.539 ±0.23 | 53.000 ±0.00 |
| **L+D+N** | 74.721 ±0.17 | 74.721 ±0.17 | 74.721 ±0.17 | 74.721 ±0.17 | 74.400 ±0.40 | 55.852 ±0.82 | 55.852 ±0.82 | 55.852 ±0.82 | 55.852 ±0.82 | 54.600 ±1.37 |

TABLE IV
VIDEO EMBEDDING RESULT: DIFFERENT USER-EMBEDDING WISE GL-SCORE IMPACT ON THE RESULTS WHEN *(A)* A SINGLE FRAME IS EXTRACTED FROM EVERY SINGLE SCENE AND (*B*)FRAMES ARE EXTRACTED AFTER APPLYING SCENE AND YOLOV5 OBJECT DETECTION METHOD

| *Movies used in generating USER's Embedding* | **With GL-score** | | | | | **Without GL-score** | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | **P** | **R** | **F1 score** | **Acc.** | **Wt. Avg. Acc.** | **P** | **R** | **F1 score** | **Acc.** | **Wt. Avg. Acc.** |
| ***(A) Video Frames: [Frame from each single scene]*** | | | | | | | | | | |
| **L** | **74.455** ±0.25 | **74.455** ±0.25 | **74.455** ±0.25 | **74.455** ±0.25 | **74.600** ±0.49 | **62.277** ±0.66 | **62.277** ±0.66 | **62.277** ±0.66 | **62.277** ±0.66 | **61.800** ±0.75 |
| **L+D** | 74.250 ±0.20 | 74.250 ±0.20 | 74.250 ±0.20 | 74.250 ±0.20 | 74.000 ±0.00 | 55.275 ±0.39 | 55.275 ±0.39 | 55.275 ±0.39 | 55.275 ±0.39 | 53.200 ±0.75 |
| **L+N** | 74.427 ±0.18 | 74.427 ±0.18 | 74.427 ±0.18 | 74.427 ±0.18 | 72.400 ±0.40 | 55.433 ±0.32 | 55.433 ±0.32 | 55.433 ±0.32 | 55.433 ±0.32 | 53.800 ±0.40 |
| **L+D+N** | 73.743 ±0.25 | 73.743 ±0.25 | 73.743 ±0.25 | 73.743 ±0.25 | 73.600 ±0.49 | 59.409 ±0.42 | 59.409 ±0.42 | 59.409 ±0.42 | 59.409 ±0.42 | 59.200 ±0.40 |
| ***(B) Video Frames: [Frame from Scene + Object Detection Method]*** | | | | | | | | | | |
| **L** | 73.738 ±0.35 | 73.738 ±0.35 | 73.738 ±0.35 | 73.738 ±0.35 | 73.600 ±0.49 | **62.612** ±0.40 | **62.612** ±0.40 | **62.612** ±0.40 | **62.612** ±0.40 | **62.200** ±0.40 |
| **L+D** | **73.925** ±0.25 | **73.925** ±0.25 | **73.925** ±0.25 | **73.925** ±0.25 | **73.800** ±0.40 | 54.669 ±0.42 | 54.669 ±0.42 | 54.669 ±0.42 | 54.669 ±0.42 | 53.000 ±0.63 |
| **L+N** | 73.580 ±0.37 | 73.580 ±0.37 | 73.580 ±0.37 | 73.580 ±0.37 | 73.600 ±0.49 | 54.767 ±0.19 | 54.767 ±0.19 | 54.767 ±0.19 | 54.767 ±0.19 | 53.000 ±0.00 |
| **L+D+N** | 72.868 ±1.86 | 72.868 ±1.86 | 72.868 ±1.86 | 72.868 ±1.86 | 73.000 ±2.00 | 60.549 ±0.44 | 60.549 ±0.44 | 60.549 ±0.44 | 60.549 ±0.44 | 60.200 ±0.40 |

has opened the corner cases to cover in building an Indian language-based movie recommendation system. Developing the Recommendation system by covering all the corner cases found in this investigation, understanding the movie's overall emotions from the facial expressions, and speech present in the trailer, and comparing the results with the state-of-the-art would be our primary objective in future work.

## REFERENCES


[1] H.-G. Kim, G. Y. Kim, and J. Y. Kim, "Music recommendation system using human activity recognition from accelerometer data," *IEEE Transactions on Consumer Electronics*, vol. 65, no. 3, pp. 349–358, 2019.

[2] S. Nayak, A. Routray, M. Sarma, and S. Uttarkabat, "Gnn based embedded framework for consumer affect recognition using thermal facial rois," *IEEE Consumer Electronics Magazine*, 2022.

[3] G. A. Prabhakar, B. Basel, A. Dutta, and C. V. R. Rao, "Multichannel cnn-blstm architecture for speech emotion recognition system by fusion of magnitude and phase spectral features using dcca for consumer applications," *IEEE Transactions on Consumer Electronics*, 2023.

[4] R. Chatterjee, S. Mazumdar, R. S. Sherratt, R. Halder, T. Maitra, and D. Giri, "Real-time speech emotion analysis for smart home assistants," *IEEE Transactions on Consumer Electronics*, vol. 67, no. 1, pp. 68–76, 2021.

[5] P. Agarwal, R. Verma, and A. Majumdar, "Indian regional movie dataset for recommender systems," *arXiv preprint arXiv:1801.02203*, 2018.

[6] R. Lavanya and B. Bharathi, "Movie recommendation system to solve data sparsity using collaborative filtering approach," *Transactions on Asian and Low-Resource Language Information Processing*, vol. 20, no. 5, pp. 1–14, 2021.

[7] G. Behera and N. Nain, "Collaborative filtering with temporal features for movie recommendation system," *Procedia Computer Science*, vol. 218, pp. 1366–1373, 2023.

[8] M. J. Pazzani, "A framework for collaborative, content-based and demographic filtering," *Artificial intelligence review*, vol. 13, pp. 393–408, 1999.

[9] S. Sridhar, D. Dhanasekaran, and G. Latha, "Content-based movie recommendation system using mbo with dbn." *Intelligent Automation & Soft Computing*, vol. 35, no. 3, 2023.

[10] S. Reddy, S. Nalluri, S. Kunisetti, S. Ashok, and B. Venkatesh, "Content-based movie recommendation system using genre correlation," in *Smart Intelligent Computing and Applications: Proceedings of the Second International Conference on SCI 2018, Volume 2*. Springer, 2019, pp. 391–397.

[11] N. Pradeep, K. Rao Mangalore, B. Rajpal, N. Prasad, and R. Shastri, "Content based movie recommendation system," *International journal of research in industrial engineering*, vol. 9, no. 4, pp. 337–348, 2020.

[12] S. Hwang and E. Park, "Movie recommendation systems using actor-based matrix computations in south korea," *IEEE Transactions on*

TABLE V
VIDEO EMBEDDING RESULT: DIFFERENT USER-EMBEDDING WISE GL-SCORE IMPACT ON THE RESULTS WHEN FRAMES ARE EXTRACTED IN EQUAL INTERVALS AND EMBEDDINGS ARE GENERATED USING RESNET-50

| Movies used in generating USER's Embedding | Wth GL-score | | | | | Without GL-score | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | P | R | F1 score | Acc. | Wt. Avg. Acc. | P | R | F1 score | Acc. | Wt. Avg. Acc. |
| L | 72.453 ±0.23 | 72.453 ±0.23 | 72.453 ±0.23 | 72.453 ±0.23 | 72.600 ±0.40 | 54.234 ±0.34 | 54.234 ±0.34 | 54.234 ±0.34 | 54.234 ±0.34 | 53.600 ±0.42 |
| L+D | 71.703 ±0.17 | 71.703 ±0.17 | 71.703 ±0.17 | 71.703 ±0.17 | 71.800 ±0.21 | 54.534 ±0.53 | 54.534 ±0.53 | 54.534 ±0.53 | 54.534 ±0.53 | 53.400 ±0.56 |
| L+N | 72.117 ±0.25 | 72.117 ±0.25 | 72.117 ±0.25 | 72.117 ±0.25 | 72.000 ±0.00 | 54.000 ±0.34 | 54.000 ±0.34 | 54.000 ±0.34 | 54.000 ±0.34 | 53.000 ±0.63 |
| L+D+N | 72.527 ±0.19 | 72.527 ±0.19 | 72.527 ±0.19 | 72.527 ±0.19 | 72.600 ±0.23 | 54.646 ±0.41 | 54.646 ±0.41 | 54.646 ±0.41 | 54.646 ±0.41 | 53.600 ±0.40 |

TABLE VI
PROPOSED MODEL(PM) VS. ARTIFICIAL NEURAL NETWORK(ANN) MODEL PERFORMANCE FOR EQUAL INTERVAL BASED FRAMES AND WAV2VEC+GENRE BASED AUDIO EMBEDDING. [↑: PERCENTAGE INCREASED IN PERFORMANCE OF PM COMPARED WITH ANN]

| USER's Embedding | Model | Video | Audio | Audio+Video |
|---|---|---|---|---|
| | | Wt. Avg. | Wt. Avg. | Wt. Avg. |
| L | PM | **74.8** (3.4% ↑) | **79** (8% ↑) | **77** (5% ↑) |
| | NN | 71.4 | 71 | 72 |
| L+D | PM | **74** (4% ↑) | **76.2** (8.2% ↑) | **75** (4% ↑) |
| | NN | 70 | 68 | 71 |
| L+N | PM | **74** (3.2% ↑) | **76.2** (8.4% ↑) | **75** (4.8% ↑) |
| | NN | 70.8 | 67.8 | 70.2 |
| L+D+N | PM | **73** (2.2% ↑) | **80** (11% ↑) | **75** (4.2% ↑) |
| | NN | 70.8 | 69 | 70.8 |

*Computational Social Systems*, vol. 9, no. 5, pp. 1387–1393, 2021.

[13] S. Kumar, K. De, and P. P. Roy, "Movie recommendation system using sentiment analysis from microblogging data," *IEEE Transactions on Computational Social Systems*, vol. 7, no. 4, pp. 915–923, 2020.

[14] S. S. Choudhury, S. N. Mohanty, and A. K. Jagadev, "Multimodal trust based recommender system with machine learning approaches for movie recommendation," *International Journal of Information Technology*, vol. 13, pp. 475–482, 2021.

[15] Y. Mu and Y. Wu, "Multimodal movie recommendation system using deep learning," *Mathematics*, vol. 11, no. 4, p. 895, 2023.

[16] Y. Deldjoo, "Enhancing video recommendation using multimedia content," *Special Topics in Information Technology*, pp. 77–89, 2020.

[17] B. Yang, T. Mei, X.-S. Hua, L. Yang, S.-Q. Yang, and M. Li, "Online video recommendation based on multimodal fusion and relevance feedback," in *Proceedings of the 6th ACM international conference on Image and video retrieval*, 2007, pp. 73–80.

[18] A. Kaklauskas, R. Gudauskas, M. Kozlovas, L. Peciure, N. Lepkova, J. Cerkauskas, and A. Banaitis, "An affect-based multimodal video recommendation system," *Studies in Informatics and Control*, vol. 25, no. 1, p. 6, 2016.

[19] S. Pingali, P. Mondal, D. Chakder, S. Saha, and A. Ghosh, "Towards developing a multi-modal video recommendation system," in *2022 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2022, pp. 1–8.

[20] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE transactions on neural networks*, vol. 20, no. 1, pp. 61–80, 2008.

[21] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio *et al.*, "Graph attention networks," *stat*, vol. 1050, no. 20, pp. 10–48 550, 2017.

[22] D. Chakder, P. Mondal, S. Raj, S. Saha, A. Ghosh, and N. Onoe, "Graph network based approaches for multi-modal movie recommendation system," in *2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2022, pp. 409–414.

[23] S. Raj, P. Mondal, D. Chakder, S. Saha, and N. Onoe, "A multimodal multi-task based approach for movie recommendation," in *2023 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2023, pp. 1–8.

[24] X. N. Lam, T. Vu, T. D. Le, and A. D. Duong, "Addressing cold-start problem in recommendation systems," in *Proceedings of the 2nd international conference on Ubiquitous information management and communication*, 2008, pp. 208–211.

[25] C. Feng, Z. Liu, S. Lin, and T. Q. Quek, "Attention-based graph convolutional network for recommendation system," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 7560–7564.

[26] P. Mondal, D. Chakder, S. Raj, S. Saha, and N. Onoe, "Graph convolutional neural network for multimodal movie recommendation," in *Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing*, 2023, pp. 1633–1640.

[27] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[28] S. Katkamr, A. Atikam, P. Mahesh, M. Chatre, S. S. Kumar, and G. Sakthidharan, "Content-based movie recommendation system and sentimental analysis using ml," in *2023 7th International Conference on Intelligent Computing and Control Systems (ICICCS)*. IEEE, 2023, pp. 198–201.

[29] Y. Deldjoo, M. Elahi, P. Cremonesi, F. Garzotto, P. Piazzolla, and M. Quadrana, "Content-based video recommendation system based on stylistic visual features," *Journal on Data Semantics*, vol. 5, pp. 99–113, 2016.

[30] G. Bertasius, H. Wang, and L. Torresani, "Is space-time attention all you need for video understanding?" in *ICML*, vol. 2, no. 3, 2021, p. 4.

[31] X. Zhu, S. Lyu, X. Wang, and Q. Zhao, "Tph-yolov5: Improved yolov5 based on transformer prediction head for object detection on drone-captured scenarios," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 2778–2788.

[32] G. Jocher, "YOLOv5 by Ultralytics," 5 2020. [Online]. Available: https://github.com/ultralytics/yolov5

[33] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in neural information processing systems*, vol. 33, pp. 12 449–12 460, 2020.

[34] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[35] Z. Zhang and M. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," *Advances in neural information processing systems*, vol. 31, 2018.

[36] D. Berrar, "Cross-validation." 2019.

[37] Q. Bao, Y. Liu, B. Gang, W. Yang, and Q. Liao, "S 2 net: Shadow mask-based semantic-aware network for single-image shadow removal," *IEEE Transactions on Consumer Electronics*, vol. 68, no. 3, pp. 209–220, 2022.

**Prabir Mondal** is presently engaged in the pursuit of his doctoral degree at the esteemed National Institute of Technology Patna, specializing in the field of Computer Science and Engineering. His research area of focus revolves around the development of a multimodal recommendation system, a topic of significant contemporary relevance in the realm of information technology. Prior to his doctoral endeavors, Mr. Mondal successfully earned his Bachelor of Technology degree in Information Technology from the prestigious Government College of Engineering and Ceramic Technology, thereby demonstrating a strong academic foundation in his chosen field.

**Pulkit Kapoor** received his M.Tech degree in Mechatronics from the Indian Institute of Technology, Patna where his M.Tech thesis primarily focused on the development of a Multi-modal recommendation System. Prior to his M.Tech, he earned his B.Tech degree in Mechanical Engineering from the Inderprasth Engineering College. Currently, Pulkit Kapoor has been serving as a Data Scientist in the industry. In his current role, he is actively involved in the deployment of advanced Machine Learning models and the development of a Research Graph.

**Siddharth Singh** is currently pursuing his B.Tech degree with the Department of Electrical Engineering, Indian Institute of Technology, Jodhpur, India. His research interests include Neural Networks, Computer Vision, Recommendation Systems, Image Processing, and signal processing.

**Sriparna Saha** received M.Tech and Ph.D. degrees in computer science from Indian Statistical Institute Kolkata, Kolkata, India. She is currently an Associate Professor with the Department of Computer Science and Engineering, Indian Institute of Technology Patna, India. From September 2009 to June 2010, she was a Postdoctoral Research Fellow at the University of Heidelberg, Germany. From September 2010 to January 2011, she was a Postdoctoral Research Fellow with the Department of Information Engineering and Computer Science at the University of Trento, Italy. Her current research interests include text mining pattern recognition, natural language processing, multiobjective optimization, and biomedical information extraction. She has more than 400 publications with 7000 citations.

**Jyoti Prakash Singh** is an Associate Professor in the Department of Computer Science and Engineering at the National Institute of Technology Patna, India. He has published over 60 international journal publications and more than 60 international conference proceedings. He was involved as an investigator in the MeitY-sponsored project to develop algorithms for spam calls/fake calls in a telephonic conversation. Dr. Singh has been recognized in the "World ranking of top 2% scientists" in the area of "Artificial Intelligence" (Year 2021), according to a survey given by Stanford University, USA. His research interests focus on social media mining, deep learning, information security, and speech processing. He is a senior member of IEEE and ACM. Dr. Singh is the Associate Editor of Network : Computation in Neural Systems and International Journal of Electronic Government Research. He was awarded the S4DS Data Scientist (Academia) Award by the Society for Data Science in 2020.

**Amit Kumar Singh** is an Associate Professor in the Computer Science and Engineering Department at the National Institute of Technology Patna, Bihar, India. He has authored over 150 peer-reviewed journals, conference publications, and book chapters. Dr. Singh has been recognized in the "World ranking of top 2% scientists" in the area of "Biomedical Research" (Year 2019) and "Artificial Intelligence & Image Processing" (Year 2020), according to a survey given by Stanford University, USA. Dr. Singh is the Associate Editor of IEEE Transactions on Industrial Informatics, IEEE Transactions on Computational Social Systems, IEEE Journal of Biomedical and Health Informatics, Engineering Applications of Artificial Intelligence etc. His research interests include multimedia data hiding, image processing, biometrics, and cryptography. Contact him at amit.singh@nitp.ac.in.