Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Movie Recommendation and Sentiment Analysis using Deep Learning Algorithms

Dr. M. Sunitha[1], Dr.T.Adilakshmi [2], V Siri Vaishnavi[3]
[1]*Department of Computer Science and Engineering, Vasavi College of Engineering, India*
[2]*Department of Computer Science and Engineering, Vasavi College of Engineering, India*
[3] *Department of Computer Science and Engineering, Vasavi College of Engineering, India.*

ARTICLE INFO

Received: 30 Dec 2024

Revised: 12 Feb 2025

Accepted: 26 Feb 2025

ABSTRACT

Data available on the internet is growing constantly. This huge amount of data makes it harder for the users to get useful information. Comments and reviews of movies are given by many users, and recommendation systems makes it easier to find useful content, which is fast and relevant for users. Movies with more positive reviews and comments are usually chosen by everyone. The sentiment behind user reviews is useful to know whether a movie is worth watching. This paper outlines an approach for a movie recommendation system that uses cosine similarity technique to recommend similar movies to users based on the movie title , genre, director, actor they choose or search . Deep learning algorithms, GRU (Gated Recurrent unit), LSTM (long short-term memory), RNN (Recurrent neural network) and BI-LSTM (bi-directional long short-term memory), are trained and tested to classify user comments taken from YouTube into positive, negative, and neutral sentiments. Accuracy, F1-score, precision, recall measures are utilized to assess the model from every perspective. After comparing four algorithms, as BI-LSTM outperformed other algorithms with an accuracy, recall, F1 score and precision, of 98.58%, 97.99% 98.26% and 98.54% respectively, Bi-LSTM (bi-directional long short-term memory) is used for performing sentiment analysis on the movie reviews dataset to understand users sentiment.

Keywords: Recurrent neural networks (RNN), Long short term memory(LSTM), Gated Recurrent unit(GRU), bi-directional long short term memory(Bi-LSTM), Sentiment analysis, recommendation

INTRODUCTION

The Internet is always growing. Users can now rate, comment and post reviews for any service or product which is available online, and this sheer amount of content available might make it difficult to get the information you need quickly. Thankfully, recommendation systems make this process easier by filtering out useful data for users and providing tailored recommendations according to user preferences [1].

These recommendation systems are now a crucial component of many different businesses, such as retail, finance, entertainment, and e-commerce. By evaluating user data they provide automated recommendations that are unique to different users [11]. Recommendation systems most commonly use three techniques: hybrid filtering, CF (collaborative filtering), CBF (content-based filtering). CBF is a method used for recommendation which makes product recommendations for users according to the characteristics of products they have already used. According to this method, if a user has a preference for one item, they will also prefer items with similar attributes. Collaborative filtering is a recommendation method that overcomes the limits of CBF by combining a user's previous preferences with the behaviours and preferences of users with similar interests [6]. Unlike CBF, CF focuses on user interactions and forms patterns to make recommendations. Many recommendation systems use hybrid filtering techniques which combine CF with content-based filtering to improve accuracy and to overcome any limitations [16] .

Users usually choose movies which have the most positive reviews and comments and tend to seek movies with higher ratings rather than movies having negative reviews or lower ratings. Movies popularity also depends on users' opinions. Users can make decisions to choose a movie they want by considering positive reviews and ignoring negative or misleading comments. This can be done by sentiment analysis. Opinion mining, another name for sentiment analysis, represents a significant area of study within natural language processing [19]. Sentiment analysis uses NLP to classify users' opinions and words into positive and negative fields, which helps in improving user experience [18].

Sentiment analysis through opinion mining can provide valuable information regarding the opinions and sentiments of the general public. This is vital for corporations, governments, and organisations that desire to formulate plans, forecast market trends, and effectively address issues of public opinion. Developing efficient procedures and techniques that enable us to comprehend and make use of the enormous volume of online public opinion data is the aim of sentiment analysis mining. It seeks to accurately gauge the distribution and sentiment of opinions as well as identify the most popular subjects. [21].

This paper introduces a system that recommends movies to users according to their preferences and analyses the sentiment of the reviews to categorise them as positive, negative or neutral. The system's movie recommendation part utilizes cosine similarity, which assesses the similarity among items or users. Sentiment analysis is performed by comparing the deep learning algorithms LSTM, RNN, Bi-LSTM and GRU.

The objective of this paper is to tackle with huge amount of data that is available to users and filter out useful data, recommending movies to users according to their preferences and applying sentiment analysis on the reviews to increase user experience.

This paper consists of the following structure: The introduction section introduces the proposal, context, motivation and objectives of this study. Literature review analyses different papers from different authors and research gaps among them for further study. The proposed model discusses the datasets used and pre-processing techniques applied. It also focuses on understanding this paper recommendation model and sentiment analysis. Results are focused on the paper's findings and contributions. The report ends with a conclusion which summarises the entire study for better understanding.

LITERATURE REVIEW

An effective recommendation model which is hybrid and combines CBF and CF is controlled by the set of rankings generated by the self-organising map, an established artificial neural network technique for unsupervised learning, was proposed in the paper discussed by [20]. Collaborative filtering creates user-item interaction scores by matrix factorising using SVD. To measure item attributes and cosine similarity for suggestions, content-based filtering uses TF-IDF. CF Based on SOM, it uses SVD to enhance the cluster-specific suggestion and self -organising maps (SOM) to group movies according to demographic characteristics and genres. The MovieLens 100k dataset, which includes 100,000 user reviews for 1,682 films, was used. RMSE, precision, and recall at k and f1 score were employed as evaluation metrics. In terms of precision, recall, and RMSE, the hybrid model performed better than conventional CF and CBF.

Collaborative filtering was combined with K-means clustering in their previous paper by [4] . K-means is used to group items (movies) by genre attributes, and users are classified according to the genres and items they are interested in seeing. Segmented user profiles were created using user demographics like age and gender. Next, to suggest products to users, the CFtechnique is used on the group in which user belongs.[4] Compared to K-Means, SOM-based CF demonstrated superior clustering efficiency and recommendation accuracy.

A system that combines sentiment analysis and movie recommendation, driven by sophisticated frameworks and algorithms, was proposed by [8]. Cosine similarity, based on angular similarity of feature vectors, was used to suggest movies to users according to their interests. For the purpose of this study, three datasets were employed. One dataset for sentiment analysis, and the other two are for movie recommendations. Reviews.txt is used for sentiment analysis, while "tmdb_5000_movies.csv" and "tbmd_5000_credits.csv" are used for movie recommendations. Machine learning algorithms SVM and NB were compared to perform sentiment analysis on reviews. In order to maximise performance, the two machine learning algorithms are compared using precision, recall, accuracy and F1-score metrics. (SVM) support vector machine achieved 98.63 percentage of accuracy,

2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

outperforming (NB) naïve bayes in every metric. The comparison demonstrated how well SVM handles complex datasets.

With a goal to enhance recommendations by incorporating real-time public opinion, a hybrid RS that combines CF, CBF, and sentiment analysis of tweets about movies is proposed by [17]. The author integrated sentiment analysis of tweets with hybrid RS to overcome limitations. The Movietweetings dataset with tweets of the movies from 1894 to 2017 is used and was modified, as tweets from earlier movies were difficult to find. dataset with tweets for the movies released from 2014 were considered and modified. The VADER algorithm scores the tweets based on lexicon intensity, which is normalised into a range for recommendations. A linear system framework is used to optimize the weights of various metadata, including genre and director, by combining metadata-based similarity with collaborative filtering based on social graphs. To make the system robust and increase recommendation accuracy, hybrid similarity and sentiment similarity are combined, and user tweets are used to calculate sentiment-driven scores. The use of micro-blogging data was useful to keep track of real time public opinions and sentiments.

Challenges such as sparsity and cold start issues are addressed by [7]. Advanced approaches like collaborative filtering, matrix factorisation, and similarity-based methods are used to provide accurate user recommendations. The machine learning model XGBoost, a gradient boosting model, is used. A KNNBaseline, which uses k-nearest neighbours to find similarities between users or items for creating customised predictions, and a Baseline Predictor, which considers average movie ratings and user-specific biases are used . SVD (Singular Value Decomposition), a matrix factorisation method, which reduces the user and item engagment matrix to discover unused features is used by the author .SVD++, an extension of SVD, includes implicit feedback like user engagement data (e.g., browsing or watching history). SVD++ performed great, with a 1.0675 RMSE, showing that it used implicit feedback the most effectively. Because XGBoost could not handle implicit feedback, it performed decently but not as well as SVD++. KNN and other baseline models were accurate, but they were mostly based on explicit rating data.

Determining a movie's success in its early stages is challenging. [14] Introduced a new framework to predict a movie's popularity and target audience by incorporating various machine learning algorithms and recommendation systems. A fresh structure by using a CBF recommendation system which uses the actor, director and genre of a movie to find similarities with the upcoming movies is provided. A conventional neural network (CNN) is used for the movie popularity prediction module, which divides movies into six different categories, such as 'super duper hit' and 'flop'. Target audience prediction is done by using fuzzy c-means clustering and cosine similarity by dividing users into different age-specific categories: junior, teenage, middle-aged, and senior. The content-based model uses the tmdb_5000_movies and tmdb_5000_credits datasets, which are open to everyone. This movie popularity model uses the IMDb rating dataset. The CNN model improves earlier models and gives an accuracy of 96.8. The content-based model effectively adds input data to the CNN model. Age-specific categorised data is used to understand the movie market.

Incorporating artificial neural networks in a movie recommendation hybrid system built on CBF and CF is done by [15] . The issues like data sparsity and scalability are addressed by using improved SVD, content-driven KNN, IKSOM clustering, and K-means++. A matrix factorisation, (SVD) Singular Value Decomposition approach, is used to decrease the dimensionality and give user,item relationships. Cosine similarity is used to identify patterns for recommending similar movies to users based on attributes like genre and year of release. This content-driven KNN overcomes cold start problems. Improved Kohonen Self-Organising Map (IKSOM) with EISEN Distance reduces overlapping problems and improves neighbour detection, which helps in increasing accuracy. K means ++ is used to produce high-quality classification of movies. Evaluation metrics used here are RMSE and mean absolute error for predicting accuracy, which are 0.414 and 0.279, respectively. Recall, F1-score and Precision are used to measure the relevance of recommendations, which is 93.09%, 93.82%, 93.09%,93.82%,92.44% . It shows the system is able to give accurate recommendations.

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

PROPOSED MODEL

Dataset

This paper uses three databases. We use one database for the recommendation model and two databases for sentiment analysis. The movieReviews.csv and movies.csv databases contain around 150,000 reviews of 3,015 movies. These two databases have a common column for movie_id (Rotten_tomatoes_link). Data for these databases was web scraped from the Rotten Tomatoes website. The sentiment_comments.csv database is used for calculating and comparing the best deep learning algorithm between RNN, GRU, LSTM and BI-LSTM. This database consists of around 18,410 comments on various YouTube videos. This database contains a column for the sentiment of the video comments where 0 indicates negative, 1 represents neutral, and 2 represents positive sentiments. The movies.csv database is used for the recommendation model where top most similar movies according to genre ,director etc asked by the user is recommended to them, and movieReviews.csv is used for sentiment analysis after comparing four deep learning algorithms LSTM, RNN, BI-LSTM and GRU. The best-performing algorithm among the four is chosen to perform sentiment analysis with. 70 % of these databases are for training, and rest for testing.

(Table 1) Shows Dataset sample of movie_reviews.csv with 5 rows which is used for sentiment analysis .**(**

Table 2**)** shows Dataset sample of sentiment_comments.csv with 5 rows which contain several Youtube comments and is used for classifying sentiments using deep learning algorithms. The best performing algorithm is then used for sentiment analysis.

Rotten_tomatoes_link	critic_name	publisher_name	review_type	review_date	review_content
878835	Justin Chang	Variety	Fresh	25-01-2010	Like Holofcener's previous pictures, Please Give derives ...
878835	Kirk Honeycutt	Hollywood Reporter	Rotten	25-01-2010	A muted, almost Rohmer-like moral tale that doesn't quite dive...
878835	Tim Grierson	Screen International	Fresh	25-01-2010	Keener, Peet and Hall all shine as women plagued with self-doubt....
878835	Andrew O'Hehir	Salon.com	Fresh	28-01-2010	An edgy, somber, beautifully written Manhattan fable of guilt,....
878835	Erik Childress	eFilmCritic.com	Fresh	31-01-2010	Holofcener always gives us more to chew on than originally meets ....

Table 1. 5 rows of Movie review Dataset used for sentiment analysis

sno	Video ID	Comment	Likes	Sentiment
0	18fwz9Itbvo	When he says he's a 400 year old beer wizard, I believe him.	5092	1
1	18fwz9Itbvo	This series is exactly what career day should've been, such interest...	765	2
2	18fwz9Itbvo	It's far less energy intense than smelting and remanufacturing cans.	693	1
3	18fwz9Itbvo	How does every person WIRED brings on this show act like they've	263	0

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

		been in front of cameras 5,000 times?		
4	18fwz9Itbvo	idk who is picking these people, but give them a raise	8073	1

Table 2. 5 rows of Sentiment comments Dataset containing comments of various videos.

Data Pre-Processing

To improve consistency and to make text data more uniform, normalisation is performed. Text normalisation makes it easier for NLP models to process and understand data. Here we applied text normalisation techniques before performing sentiment analysis on movie reviews and before training and testing deep learning techniques on YouTube comments to make data easier to handle and comprehend.

All the texts are converted to lowercase, and new line characters within the text are eliminated. Subsequently, punctuation marks are removed, along with hashtags and references that complicate the text. Spaces

in text can prolong the process of sentiment analysis and recommendation; therefore, to save time, multiple spaces in the text are removed. Additionally, special characters in text increase its complexity, so these are removed as well. Common words such as a, the, and is in English texts are often meaningless and need to be filtered out. The process of stop word removal is undertaken to facilitate easier text processing. Lemmatization is a pre-processing technique that reduces words in the text to their dictionary base forms. It takes into account the context of words and their parts of speech, thereby reducing them to their root forms. Here, we employed the WordNetLemmatizer from the NLTK package to convert the words into their basic forms for more efficient text processing.

Recommendation model

In the recommendation model, users can see the top 10 movies, which are calculated based on audience count. The top movies with the highest audience count constitute the top movies section. Users are recommended similar movies when they search for a title, genre, director, or actors of a movie that they are likely to enjoy. When users wish to watch films in the drama genre and search for such titles, the top films in the drama category are recommended to the user. These similar top movies are suggested to users using the TFIDF technique and a concept known as cosine similarity.

The recommendation model uses the movies.csv file. Initially, a target movie ID is selected from the movie list, after which all other movies, excluding the target movie, are extracted. Then, the movie information for all the movies is extracted; this consists of a list of textual descriptions (or metadata) for each movie. Finally, we append the movie information for the target movie to the movie information list for comparison.

Next, the vectorization of text is performed utilizing Term Frequency-Inverse Document Frequency technique. This technique aids in identifying unique words within a document or text by converting the text into numerical vectors . It is a technique employed in content-based analysis to ascertain the relevance of a word within a document in relation to other documents. Stop words such as "a", "the", and "is", which are common in English, are removed to facilitate easier vectorization. These numerical vectors of text assist in determining the significance of those words within that document or text. Here in the proposed model of the recommendation system, the movie information text is converted into numerical vectors using the TF-IDF technique. After converting text into vectors, a matrix factorisation technique SVD (Singular Value Decomposition) is applied to decrease the size (dimensionality) of the vectors .This helps us to speed up the process of calculating similarities and also helps in removing noise in the data. Subsequently, we compare the final vector, which represents the target movie information, with all the other movie information vectors through cosine similarity.

Cosine similarity is utilised to compare two entities, such as users or items, and ascertain how similar they are to one another. It specifically measures the angle between these two entities. **(Equation 1)** shows the equation of cosine similarity. If the angle is small, then the two entities are similar to one another, and if the angle is large, then the two entities are different from one another.

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

$$\cos\theta = \frac{P.Q}{||P||\,||Q||} \quad \text{(Equation 1)}$$

Where,

P and Q represents two vectors

||P|| represents the magnitude of vector P and ||Q|| represents the magnitude of vector Q

Here, after using TF-IDF to convert movie information lists into vectors, the angle between the target movie information vector and all other movie information vectors is calculated to determine their orientation in vector space. This angle between the vectors is used to suggest similar movies to users. Cosine similarity values range from -1 to 1, where a value of 1 indicates high similarity and -1 signifies strong opposition. The greater the number of overlapping words, particularly rare or significant ones, the higher the similarity. When a user searches for a movie, genre, or director, movies similar to the search are provided to the user. This similarity recommendation is achieved through cosine similarity. The angle between the movie requested by the user and the movies in the database is calculated, and movies that have a small angle (close to 1.0) are presented to the user.

Sentiment Analysis

RNN,GRU, LSTM and BI-LSTM are the deep learning algorithms employed for sentiment analysis. Firstly, these algorithms are trained and tested on a dataset, sentiment_comments.csv, which comprises YouTube video comments from various users across multiple videos. Each comment is assigned a sentiment number (0, 1, or 2) to classify it as negative, positive, or neutral. RNN,GRU, LSTM and Bi-LSTM are trained and tested to determine which of the two algorithms performs better. After obtaining the results, the best-performing algorithm of the four is selected to conduct sentiment analysis on the movie_reviews.csv database and assign each review its sentiment.

One type of neural network which is used to process sequential data commonly is RNN(Recurrent neural network).Unlike feedforward networks RNNs remember previous inputs using internal memory which allows them to process time series data. A type of RNN, GRU architecture was designed to fix the limitations of vanishing gradient problem of RNNs. It is computationally simpler and efficient alternative to LSTM.GRUs architecture comprises of 2 gates reset and update gate. Update gate handles how much information can be carried forward and reset gate handles how much information to forget.

Another type of RNN, LSTM is designed to remember crucial details (information) over long time periods and forget unnecessary parts. It was created to solve RNNs' vanishing gradient issue. RNN models often struggle to retain earlier information in a long sequence. LSTM, conversely, retains words for an extended period and assists in predicting the sentiment of a sentence. LSTM can remember a word from a movie review over an extended sequence and can predict its sentiment as positive, neutral, or negative. Each LSTM unit comprises of 3 gates, an output gate, an forget gate and a input gate. The forget gate determines which unused parts should be discarded

$$f_{t=\sigma(W_f \times [h_{t-1}, x_t] + b_f)} \quad \text{(Equation 2)}$$

from the previous state. **(Equation 2**) shows the equation of forget gate in a LSTM unit.

The input gate decides which data to keep in the memory for longer use, while the output gate regulates the output generated by the LSTM cell. Each gate serves a specific function and aids the LSTM unit in retaining information for an extended period. **(Equation 3)** and **(Equation 4)** show the equations of input and output gates of LSTM unit**. (Figure 1)** Represents architecture of LSTM network with 3 gates (forget, input and output gates)

$$f_{i=\sigma(W_i \times [h_{t-1}, x_t] + b_i)}$$ (Equation 3)

$$f_{o=\sigma(W_o \times [h_{t-1}, x_t] + b_o)}$$ (Equation 4)

Where,

i_t = denotes input gate

f_t = denotes forget gate

o_t = denotes output gate

w_x = weight of the respective gate (x) neurons

h_{t-1} = previous LSTM block output (at timestamp t-1)

x_t = current timestamp input

b_x = respective gates(x) bias

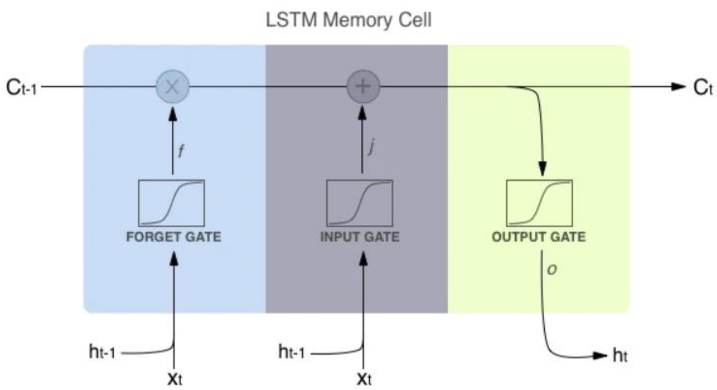

Figure 1. LSTM architecture with 3 gates (input gate, forget gate, output gate)

Bi-LSTM short for bi-directional LSTM. While LSTM reads input from left to right, Bi-LSTM processes input from two directions: right to left and left to right. Bi-LSTM operates using two LSTM networks; one processes the sequence in a forward direction by using forward pass gates, and the other processes it in reverse using backward pass gates. Then their outputs are combined. BILSTM consists of gates same as LSTM but operate twice for each step in the network.. This dual processing of data from both directions provides us with greater context regarding both the past and the future, which aids us in comprehending the data and discerning the sentiment behind the review.

Before training and testing RNN, GRU, LSTM and Bi-LSTM models, text vectorization is performed using word embeddings. Vectorization involves the conversion of text data into numerical vectors. In this process, after text tokenization and normalisation, each piece of text is transformed into integer sequences. This data is split for training and testing. Evaluations are conducted using accuracy, recall, F1 score and precision. A file named glove.6B.100d.txt is utilised, which contains a set of pre-trained word embeddings. Full form of glove is Global Vectors for Word Representation .Word embeddings are text words which represents numerical vectors . This file

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

saves training time, as we do not have to create word embeddings from scratch. Given that these embeddings are of high quality, they are extremely useful.

After training and testing four deep learning algorithms, the most effective algorithm is utilized for sentiment analysis on movie reviews, where we can get the sentiment behind every movie review, whether it is positive, neutral or negative. The top-performing model, whether it be RNN, GRU ,LSTM or a Bi-LSTM, is then loaded, and predictions are made accordingly.

Evaluation metrics

Accuracy

This metric assesses the overall correctness of a model and is particularly useful when the classes are balanced. However, when classifying YouTube comments into negative, neutral and positive sentiments, if the dataset predominantly consists of neutral sentiments, relying solely on accuracy to evaluate the model can be misleading. In that case, using recall , F1-score and precision, gives us a thorough comprehension of the model. **(Equation 5)** shows the equation of accuracy evaluation metric.

$$Acurracy = \frac{TN + TP}{TN + FP + FN + TP} \qquad \text{(Equation 5)}$$

Precision

It measures the accuracy of positive predictions. It determines how many of the instances that the model predicted as positive were actually correct. This metric is effective for avoiding false positives and is particularly useful when the false positives cost is high.**(Equation 6)** shows the equation of precision evaluation metric

.

$$Precision = \frac{TP}{TP + FP} \qquad \text{(Equation 6)}$$

Recall

Recall measures the completeness of predictions which are postive. It determines, apart from the actual positive cases, how many did the model identify correctly? It's good for avoiding false negatives and is useful when false negatives are costly.**(Equation 7)** shows equation of recall evaluation metric.

$$Recall = \frac{TP}{TP + FN} \qquad \text{(Equation 7)}$$

f1-scoref1-score combines precision and recall into a single measurement. It offers a balanced approach for the model to accurately identify positive instances, thereby minimising both false negatives and false positives. It serves as an excellent indicator when classes are imbalanced.**(Equation 8)** shows equation of f1-score evaluation metric.

$$F1\ score = 2 \times \frac{precision \times recall}{precision + recall} \qquad \text{(Equation 8)}$$

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

Where ,

TP denotes True positive

FP denotes False positive

TN denotes True negative

FN denotes False negative

In this paper we are using recall, precision, accuracy and F1-score all together. For a classification problem like sentiment analysis, it is really useful to look at the problem from all sides. Each metric tells us something slightly different, and using them all together gives us a whole picture of the model's performance.

RESULTS

After training and testing deep learning algorithms, GRU, LSTM, RNN, and BI-LSTM, on the sentiment comments database, the model is evaluated, and comparison between them is done using precision, accuracy, recall, and f1 score metrics. Here activation function for output dense layer for all the networks is Softmax which is used for classification of multiple classes (positive, negative, neutral). But for LSTM and BILSTM 2 Dense Layers are used. The hidden dense layer uses activation function Relu and the other dense layer is output layer and uses activation function Softmax. Adam optimizer is used in speeding up the training process.(**Table 3**). GRU shows an accuracy 77.03%, Precision of 73.30%, Recall of 62.46% and f1-score of 65.90%. RNN shows an accuracy 72.45%, Precision of 65.24%, Recall of 55.84%, f1-score of 58.44. LSTM shows an accuracy 97.6%, precision of 96.8%, recall of 96.9% and f1 score of 96.8%. Where as Bi-LSTM shows an accuracy 98.5%, precision of 98.5%, recall of 97.9% and f1 score of 98.2%. Graphs are plotted comparing every evaluation metric between LSTM, BI-LSTM, RNN and GRU

		Accuracy	Precision	Recall	f1-score
1	GRU	77.03	73.30	62.46	65.90
2	RNN	72.45	65.24	55.84	58.44
3	LSTM	97.67	96.81	96.91	96.86
4	Bi-LSTM	98.58	98.54	97.99	98.26

Table 3. Results obtained by deep learning algorithms with activation functions Softmax , Relu and Adam optimizer

(Error! Reference source not found.), (Error! Reference source not found.) , (Error! Reference source not found.), (Error! Reference source not found.) shows the graphs plotted to compare RNN, GRU, LSTM and BILSTM with evaluation metrics recall, F1 score , accuracy and precision with activation functions Relu, Softmax and Adam optimizer. After comparison we can observe that the bidirectional long short-term memory BI-LSTM algorithm is performing slightly better than other three algorithms in classifying YouTube comments.

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

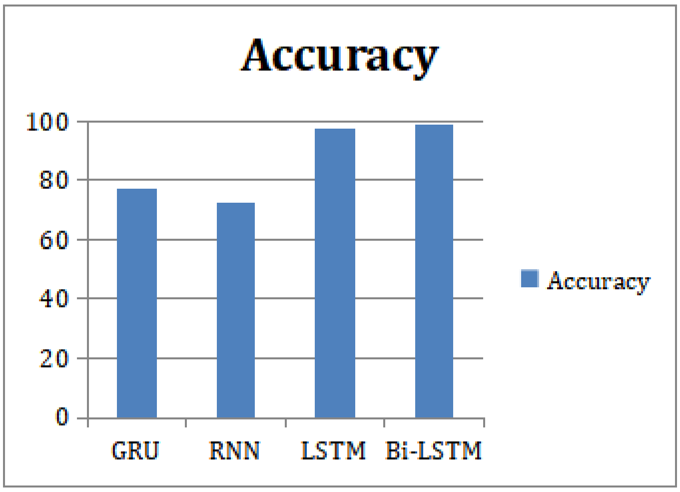

Figure 2. Accuracy obtained by deep learning algorithms

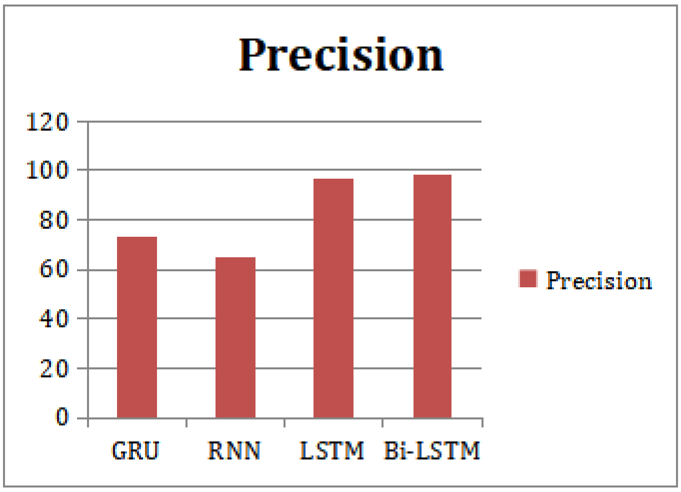

Figure 3. Precision obtained by deep learning algorithms

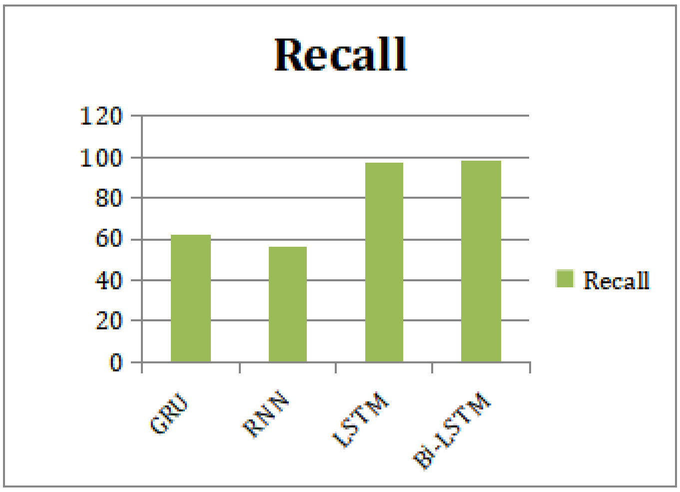

Figure4. Recall obtained by deep learning algorithms

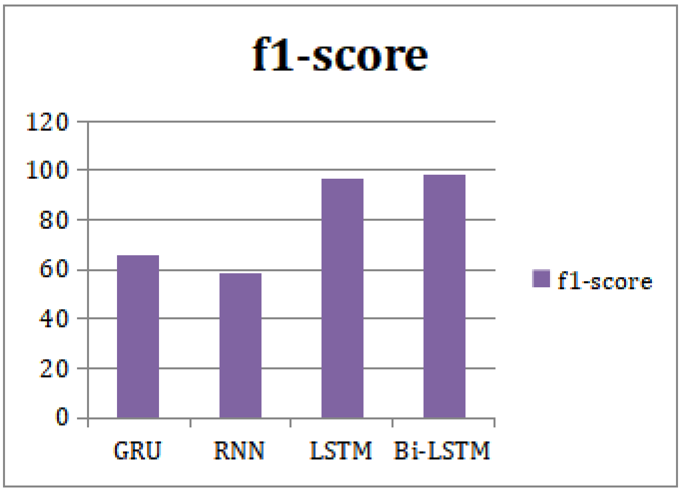

Figure 5. f1-score obtained by deep learning algorithms

(Table 4) shows the results obtained by deep learning algorithms RNN, GRU, LSTM, BILSTM .Here activation function for hidden dense layer and output sense layer for LSTM and BILSTM are Tanh and Softmax respectively .For RNN, and GRU as there is only one dense output layer only Softmax activation function is used . Adam optimizer is used for all the algorithms to increase the speed of training process.

		Accuracy	Precision	Recall	f1-score
1	GRU	77.03	73.30	62.46	65.90
2	RNN	72.45	65.24	55.84	58.44
3	LSTM	96.87	96.15	95.49	96.06
4	Bi-LSTM	98.26	97.03	97.90	97.40

Table 4. Results obtained by deep learning algorithms with activation functions Softmax , Tanh and Adam optimizer

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

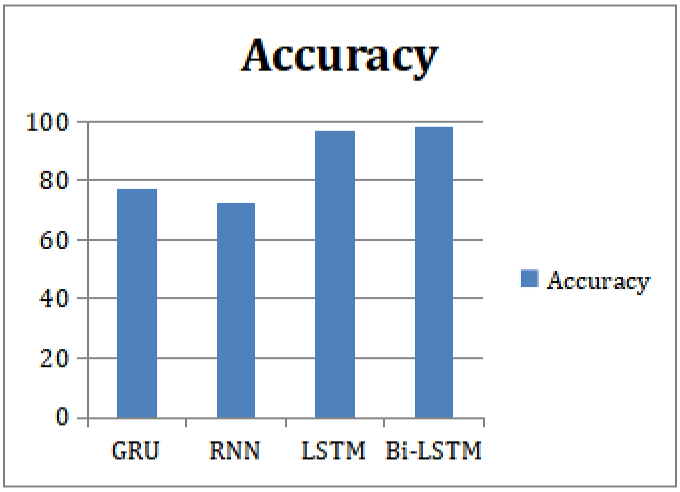

Figure 6. Accuracy obtained by deep learning algorithms

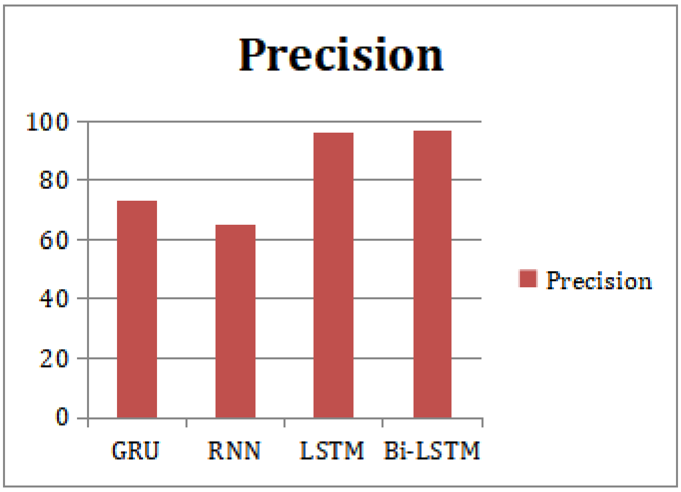

Figure 7. Precision obtained by deep learning algorithms

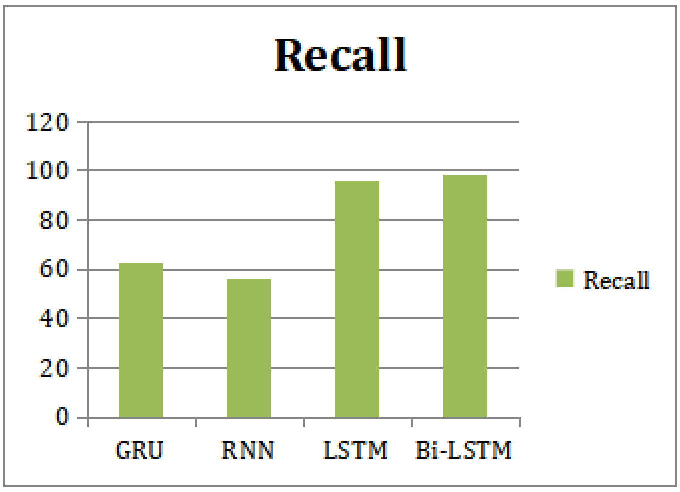

Figure 8. Recall obtained by deep learning algorithms

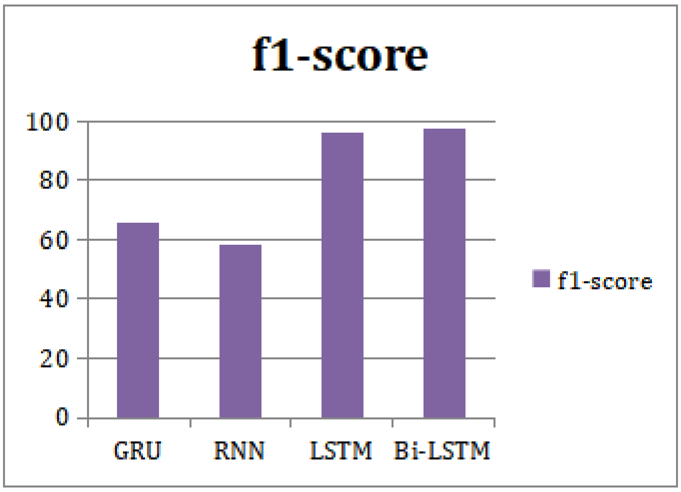

Figure 9. f1-score obtained by deep learning algorithms

(Error! Reference source not found.**)**, **(**Error! Reference source not found.**)**,**(**Error! Reference source not found.**)**, **(**Error! Reference source not found.**)** shows the graphs plotted to compare RNN, GRU, LSTM and BILSTM with evaluation metrics recall, accuracy, precision and F1 score, respectively and activation functions Tanh, softmax with adam optimizer. After comparison we can observe that BI-LSTM algorithm is performing slightly better than other three algorithms in classifying YouTube comments.

		Accuracy	Precision	Recall	f1-score
1	GRU	77.03	73.30	62.46	65.90
2	RNN	72.45	65.24	55.84	58.44
3	LSTM	95.06	96.67	96.22	96.90
4	Bi-LSTM	96.78	97.25	96.03	96.04

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

Table 5. Results obtained by deep learning algorithms with activation functions softmax , relu and rmsprop optimizer

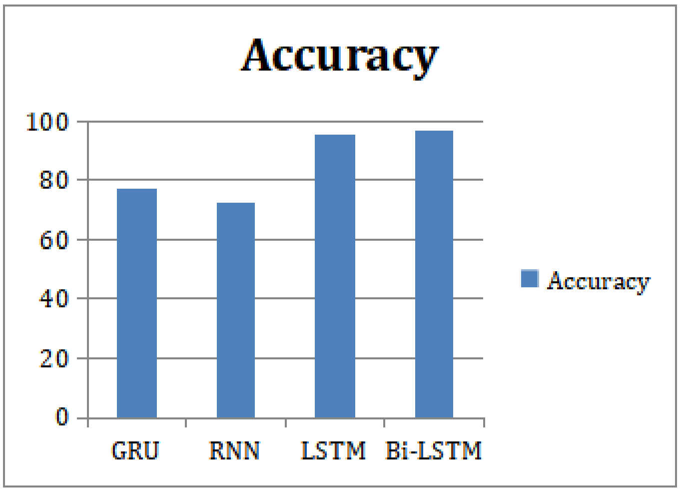

Figure 10. Accuracy obtained by deep learning algorithms

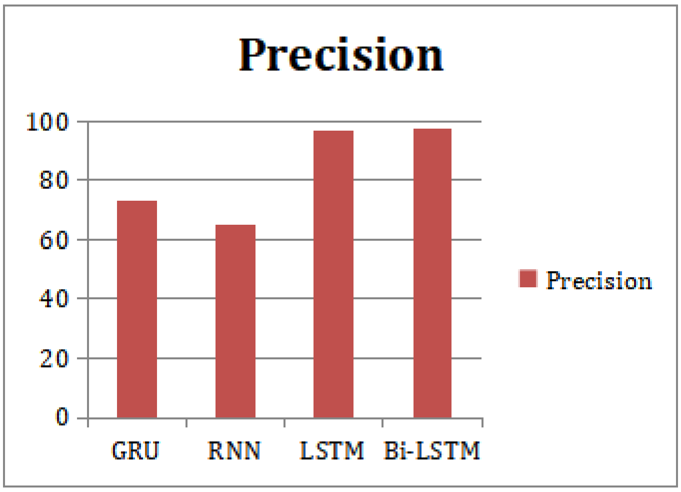

Figure 11. Precision obtained by deep learning algorithms

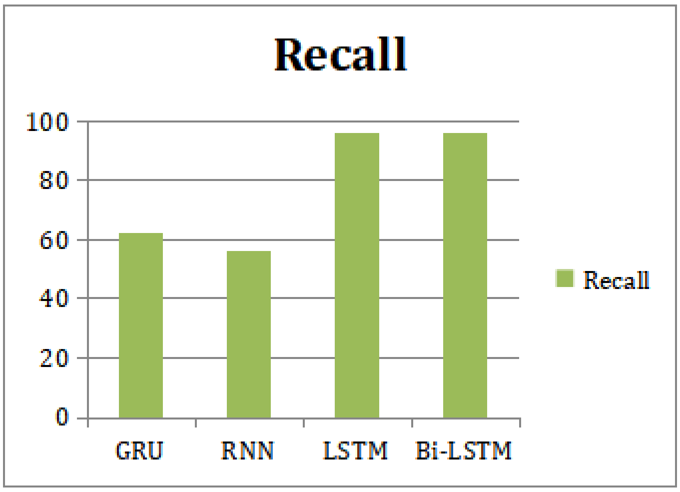

Figure 12. Recall obtained by deep learning algorithms

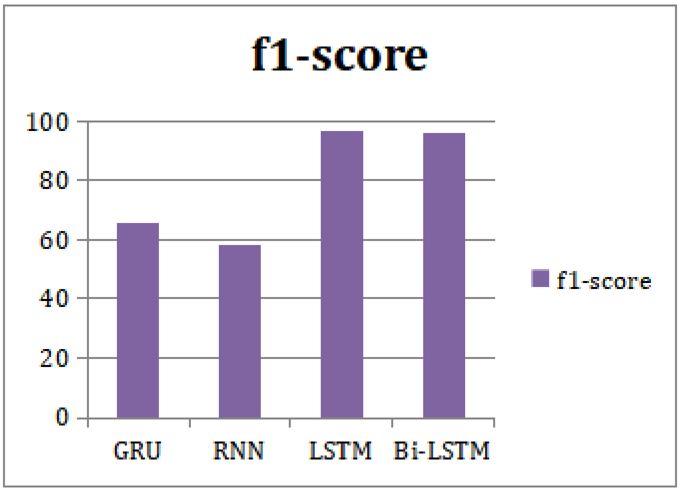

Figure 13. f1-score obtained by deep learning algorithms

(Table 5) shows the results obtained by deep learning algorithms RNN, GRU, LSTM, BILSTM .Here activation function for hidden dense layer and output sense layer for LSTM and BILSTM are relu and softmax respectively.For RNN, and GRU as there is only one dense output layer only softmax activation function is used . rmsprop optimizer is used for all the algorithms to optimize the process of training.

(Error! Reference source not found.),(Error! Reference source not found.),(Error! Reference source not found.), (Figure 14) shows the graphs plotted to compare RNN, GRU, LSTM and BILSTM with evaluation metrics recall, f1-score, accuracy and precision, respectively and activation functions Tanh, softmax with adam optimizer. After comparison we can observe that the BI-LSTM algorithm is performing slightly better than other three algorithms in classifying YouTube comments.

(Figure 14) is a pie graph plotted to show the percentage of each sentiment (positive, neutral and negative) for the movie Percy Jackson & Olympians: The Lightning Thief in the movie database with around 200 reviews present in the movie reviews dataset.

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

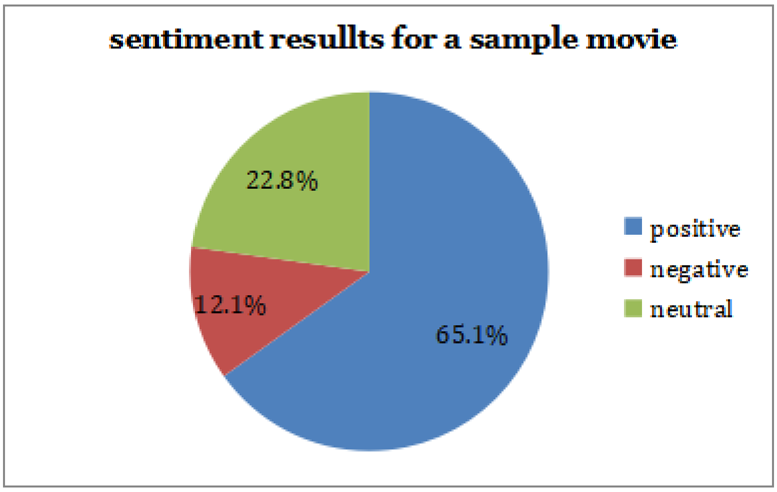

Figure 14. pie graph showing percentage of each sentiment for the movie Percy Jackson & Olympians

CONCLUSION AND FUTURE SCOPE

There are two main sections that are discussed in this paper: movie recommendation model is one part, and sentiment analysis is the other part. Both parts are discussed in detail, and some valuable conclusions are made. The recommendation model uses the cosine similarity concept to recommend best similar movies to the users according to their search. Users can search considering movie information like actors, genre, and ratings of a movie. Here, vectorization of text data is done using the concept known as TFIDF, where text is converted to a sequence of integers (vectors) and SVD(Singular value decomposition) a matrix factorisation technique is used to decrease the size of vectors which removes noise and speeds up the training process.

Sentiment analysis is done by choosing the best-performing deep learning algorithm among RNN, GRU, LSTM and BILSTM and using it to perform analysis on movie reviews to understand the sentiment behind them. The training and comparison of the four deep learning algorithms is done by performing classification on the YouTube comment database, which contains sentiments 0, 1, and 2 (positive, neutral, and negative). Here, vectorization of text data into a sequence of integers (vectors) is done using a predefined GloVe word embeddings dataset, which saves time and has quality data. The model's performance is assessed using metrics like as precision, recall, accuracy, and F1 score, and graphs are plotted for four deep learning algorithms to compare and understand the classification. Results are also compared by using different Activation functions- Tanh, Relu for hidden layers of LSTM and Bi-LSTM algorithms. For output layers of all the algorithms Softmax activation function is used with Adam and rmsprop optimizers to optimize the training process. After this, a Bi-LSTM algorithm is used for analysis on movie reviews to provide sentiment behind every movie review, as it performed better than RNN, GRU and LSTM.

Despite the system we discussed is pretty accurate, There are some limitations in it.When a user searches for a movie not present in the database, the recommendation model might not be able to recommend movies similar to the user's search. Also, after sentiment analysis, we are only able to categorise movie reviews into 3 sentiments: negative, positive and neutral. The system might not be able to recognise sarcastic reviews, and we might get a wrong classification of sentiment. For further study of movie recommendation models, we can incorporate reviews of different languages other than English for sentiment analysis for a better user experience. We can also try to incorporate generative AI into the system where, if a user asks for a movie recommendation of similar movies by describing a movie they want to watch, the system provides real-time recommendations to the user, which further increases the user experience.

REFERENCES

[1] CHANAPA CHANNARONG, CHAWISA PAOSIRIKUL, SARANYA MANEEROJ, & ATSUHIRO TAKASU. (2023). HybridBERT4Rec: A Hybrid (Content-Based Filtering and Collaborative Filtering) Recommender System Based on BERT. *IEEE Access*.

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

[2] Hossein Arabi, Vimala Balakrishnan, & Nor Liyana Mohd Shuib. (2020). A Context-Aware Personalized Hybrid Book Recommender System. *Journal of Web Engineering,*.

[3] Wei Zhao, Benyou Wang, Min Yang, Jianbo Ye, Zhou Zhao, Xiaojun Chen, et al. (2020). Leveraging Long and Short-Term Information in Content-Aware Movie Recommendation via Adversarial Training. *IEEE TRANSACTIONS ON CYBERNETICS*.

[4] Afoudi Yassine, Lazaar Mohamed, & Mohammed Al Achhab. (2021). Intelligent recommender system based on unsupervised machine learning and demographic attributes, Simul. Model. Pract. Theory.

[5] Deepjyoti roy, & mala dutta. (2022). A systematic review and research perspective on recommender systems. *Journal of Big Data*.

[6] hans Hendersen Kurniawan, William Susanto Lukman, Renaldy Fredyan, & Muhammad Amien Ibrahim. (2024). Movie Recommendation System: A Comparison of Content Based and Collaborative Filtering. *9th International Conference on Computer Science and Computational Intelligence*.

[7] M. Chenna Keshava, S. Srinivasulu, P. Narendra Reddy, & B. Dinesh Naik. (2020). Machine Learning Model for Movie Recommendation System. *International Journal of Engineering Research & Technology (IJERT), 9*(4).

[8] N Pavitha, Vithika Pungliya, Ankur Raut, Roshita Bhonsle, Atharva Purohit, Aayushi Patel, et al. (2022). Movie recommendation and sentiment analysis using machine learning. *Global Transitions Proceedings, 3*.

[9] N. K. Rao, N. P. Challa, S. S. Chakravarthi, & R. Ranjana. (2022). Movie Recommendation System using Machine Learning. *4th International Conference on Inventive Research in Computing Applications (ICIRCA)*.

[10] Prabir Mondal, Pulkit Kapoor, Siddharth Singh, Sripana singh, & Jyoti prakash singh. (2023). Task-Specific and Graph Convolutional Network based Multi-modal movie recommendation system in indian setting. *International Neural Network Society Workshop on Deep Learning Innovations and Applications*.

[11] Pradeep Kumar Singh, pijush Kanti Dutta Pramanik,, Avick Kumar Dey, & Prasenjit Choudhury. (2021). Recommender systems: an overview, research trends, and future directions. *Int. J. Business and Systems Research, 15*.

[12] QINGSHUANG YU, & WENJUAN GONG. (2020). A Deep Music Recommendation Method Based on Human Motion Analysis. *IEEE Access*.

[13] Redwane Nesmaoui, Mouad Louhichi, & Mohamed Lazaara. (2023). A Collaborative Filtering Movies Recommendation System based on Graph Neural Network. *The14th International Conference on Ambient Systems,Networks and Technologies (ANT)*.

[14] SANDIPAN SAHU, 1,RAGHVENDRA KUMAR, MOHD SHAFI PATHAN, JANA SHAFI, YOGESH KUMAR, & MUHAMMAD FAZAL IJAZ. (2022). Movie Popularity and Target Audience Prediction Using the Content-Based Recommender System. *IEEE Access*.

[15] Saurabh Sharma, & Harish Kumar Shakya. (2024). Hybrid recommendation system for movies using artificial neural network. *Expert Systems With Applications*.

[16] Sharma.S, Rana.V, & Malhotra.M. (2022). Automatic recommendation system based on hybrid filtering algorithm. *Educ Inf Technol, 27*.

[17] Sudhanshu Kumar, Kanjar De, & Kanjar De. (2020). Movie Recommendation System Using Sentiment Analysis From Microblogging Data. *IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS*.

[18] Y. Chandra, & A. Jana. (2020). Sentiment Analysis using Machine Learning and Deep Learning. *7th International Conference on Computing for Sustainable Global Development*.

[19] Yanying Mao a, Qun Liu, & Yu Zhang. (2024). Sentiment analysis methods, applications, and challenges: A systematic literature review. *Journal of King Saud University - Computer and Information Sciences, 36*.

[20] Yassine Afoudi, Mohamed Lazaar, & Mohammed Al Achhab. (2021). Hybridrecommendationsystemcombinedcontent-basedfiltering andcollaborative prediction using artificial neural network. *SimulationModellingPracticeandTheory*.

[21] Zhongnan Zhao, Wenjing Liu, & Kun Wang. (2023). Research on sentiment analysis method of opinion mining based on multi-model fusion transfer learning. *Journal of Big Data*.