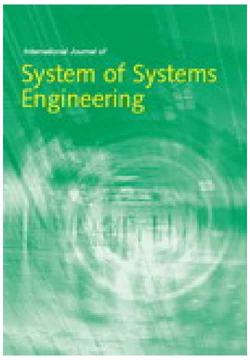


**International Journal of System of Systems Engineering**

ISSN online: 1748-068X - ISSN print: 1748-0671
https://www.inderscience.com/ijsse

# DCF-MLSTM: a deep security content-based filtering scheme using multiplicative BiLSTM for movie recommendation system

K.N. Asha, R. Rajkumar

**DOI:** 10.1504/IJSSE.2023.10053520

**Article History:**

| | |
|---|---|
| Received: | 17 March 2022 |
| Last revised: | 20 June 2022 |
| Accepted: | 21 June 2022 |
| Published online: | 16 February 2023 |

# DCF-MLSTM: a deep security content-based filtering scheme using multiplicative BiLSTM for movie recommendation system

K.N. Asha and R. Rajkumar*

School of Computing Science and Engineering,
Vellore Institute of Technology,
Vellore, 632014, Tamil Nadu, India
Email: cuashin@gmail.com
Email: vitrajkumar@gmail.com
*Corresponding author

**Abstract:** Recently, the demand for online and offline recommendation systems has increased drastically. These systems are widely used in tourism, music, and video or movie recommendations. Currently, online movie streaming applications have gained huge attention. Providing better recommendations to the user is a challenging task for these applications. The content-based filtering (CBF) recommender system is a promising technique for these systems. However, traditional systems suffer from challenges such as cold-start problems, sparsity, and scalability. Consequently, we strengthen content-based recommendation algorithms by enriching the user-related and relevant product models with effective tendencies. The majority of previous work on classifiers has been in recommendation systems. To overcome these issues, we present a deep learning model that uses a deep neural network mechanism and a multiplicative BiLSTM model. This scheme uses embedding, weight updating, and preference learning processes to improve the recommendation system's performance. The performance of the proposed approach is measured in terms of MAE, MAP, Precision, Recall, and F-measure. The comparative performance shows that the proposed approach achieves better performance when compared with state-of-art movie recommendation techniques.

**Keywords:** recommender system; multiplicative BiLSTM; deep learning security; movie recommendation system; MAE; MAP.

**Reference** to this paper should be made as follows: Asha, K.N. and Rajkumar, R. (2023) 'DCF-MLSTM: a deep security content-based filtering scheme using multiplicative BiLSTM for movie recommendation system', *Int. J. System of Systems Engineering*, Vol. 13, No. 1, pp.66–82.

**Biographical notes:** K.N. Asha works at the School of Computing Science and Engineering at the Vellore Institute of Technology in Vellore, Tamil Nadu, India. In addition to that, she has been employed as an Assistant Professor at the Department of Computer Science at the Dr. Ambedkar Institute of Technology in Bangalore, Karnataka, since 2013.

R. Rajkumar is an experienced Professor with a demonstrated history of working in the education management industry. Skilled in Data Science and Decision Making, Internet of Things, Strong education professional with a Doctor of Philosophy (PhD) and a Lifelong Learner from Vellore Institute of

Technology. He has published in various academic journals and magazines, and contributed several chapters to edited volumes. He is also the lead editor and author of referred journals and books.

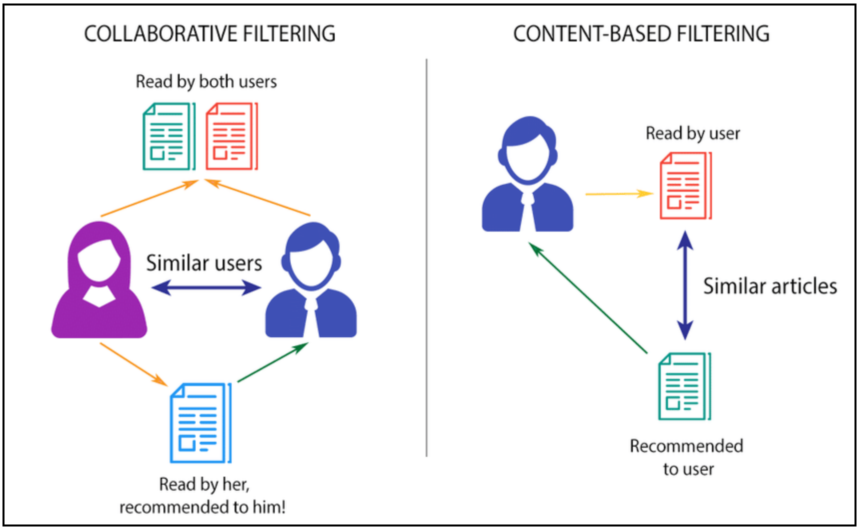
## 1 Introduction

During the last decade, the internet has become one of the prominent needs of humans due to its wide range of advantages in daily life scenarios. Due to this surge in internet usage, the amount of data on the internet-based platform has also increased. Thus, accessing fascinating information from this type of huge data has become a challenging task (Palomares et al., 2018). Recently, recommendation systems have been considered a promising solution to facilitate data access from a huge database. Various online and e-commerce applications, such as recommendations of products, music, movies, etc., utilise these recommender systems to suggest the relevant product or items to users. The best recommender systems are experienced on various online platforms such as Amazon shopping, Flipkart, Netflix, Spotify, and many more (Wei et al., 2017; Nagpal et al., 2020). The increased demand for multimedia data has led to substantial growth in online movies. It remains a challenging task for web-based multimedia applications to serve the user with movies of interest to improve the Quality of Experience of applications (Narayanasami et al., 2021). As mentioned, the recommendation system is a significant technique for these application scenarios (Kumar et al., 2017).

The current research status classifies the recommendation algorithm into three categories such as content-based recommendation technique (Son and Kim, 2017; Rutkowski et al., 2018), collaborative filtering-based recommendation technique (Raghuwanshi and Pateriya, 2019; Katarya and Verma, 2017) and hybrid recommendation technique (Yang, 2018; Paradarami et al., 2017). These techniques are widely adopted in various systems. Shu et al. (2018) presented a content-based filtering model using a convolutional neural network model to recommend the resource for e-learning systems. Rutkowski et al. (2018) presented a movie recommendation system using content-based filtering. This approach uses the Neuro-fuzzy approach for decision-making to recommend the movies. Achakulvisut et al. (2016) used a content-based filtering scheme for scientific publication recommendations. Deldjoo et al. (2016) used this approach for video recommendation by extracting and correlating visual features.

Collaborative filtering (CF) is also adopted in various real-time, online, and offline applications. Generally, this is a mechanism where past interactions between user and item are recorded and stored in a "user-item interactions matrix" to generate a suitable product recommendation. Low et al. (2018) introduced a collaborative filtering-based recommendation system using a CNN-based model. Kumar et al. (2019a) used this model for e-commerce applications to predict user ratings and recommend the product according to user's interest. Sunitha and Adilakshmi (2018) introduced user and item-based CF techniques for music recommendation. This scheme formulates the cluster of user-item ratings, and later, cluster similarity is computed to target the user for a recommendation based on ratings. Similarly, the content-based filtering scheme extracts the item features based on their content to recommend to the user based on their previous actions and explicit feedback. Several schemes have been developed based on content-based filtering for recommendation systems, such as Renuka et al. (2021) developed unsupervised content-based filtering for article recommendation (Karn et al., 2021). Wang et al. (2020) used deep learning for content-based filtering for citation recommendation (Kumar et al., 2019b). Figure 1 shows a simple collaborative and content-based filtering model for online article recommendation.

**Figure 1** A generic model of the collaborative and content-based filtering scheme (see online version for colours)



Similarly, the hybrid recommendation systems use content, collaborative-based filtering with other optimisation, and machine learning-based schemes. Chang et al. (2016) presented a two-stage hybrid model for collaborative filtering by incorporating an artificial immune system to predict the student grades. Moreover, this technique recommends the professor rating according to their courses. Cai et al. (2020) developed a hybrid recommendation system using many-objective evolutionary optimisation techniques to improve the recommendation accuracy, diversity, and coverage. Chu and Tsai (2017) developed a hybrid recommendation system for restaurant recommendations by combining content and a collaborative filtering scheme. Several studies have reported the significance of content-based filtering recommendation systems (Neffati et al., 2021). This scheme uses the known preferences of a group of users to predict unknown preferences for other users (Lazar et al., 2021).

### *1.1 Issues and challenges*

The importance of any recommender system is realised by its computation time and recommendation accuracy, but several challenges lead to degraded performance. This subsection briefly describes the challenges mentioned earlier in the previous section (Kumar et al., 2018).

*Data sparsity:* The web-based recommender systems use high-dimensional data that generates a sparse user-item interaction matrix. Generally, the cold start problem creates additional complexity because a cold start occurs when a new item is added to the system. The system cannot find similar items without interaction with other items. Similarly, if the user ratings are very small for the huge dataset, this problem is a reduced coverage (Chauhan et al., 2021).

*Scalability:* In these recommendation systems, when users grow tremendously, the traditional content-based filtering schemes suffer from the lack of computational resources.

*Gray sheep:* Gray sheep is a problem where user opinion does not agree or disagree with any group of users; thus, these users do not get benefit from a content-based filtering scheme.

*Shilling attacks:* In this scenario, the users provide many positive recommendations for their product and negative recommendations for other competing products.

Several techniques have been reported to deal with these issues during the last five years. We present a brief discussion about these techniques in Section 2.

Due to these challenging issues, our main aim is to develop a novel content-based filtering scheme to improve the performance of the online movie recommendation system (Kumar et al., 2020). We present a deep learning-based solution for a recommendation system. This scheme is developed in three phases; first, the embedding process is implemented to generate the data vector; in the next phase, the weight updating process is presented; finally, LSTM based learning model is implemented to improve the recommendation performance.

Recommenders usually consist of three elements:

- *Target generations*: Given a large pool of millions of objects, this approach is responsible for producing smaller subsets of possibilities to propose to a user.
- *Assessment processes*: Because distinct generators can generate different candidates, we need to standardise everything and provide a score for each of the subgroups' components. The scoring system is in charge of this.
- *Re-ranking structures*: After scoring is completed, the system considers other restrictions to generate the final rankings.

Multidisciplinary screening based on memory: This method involves memorising the usability testing matrix and how a user reacts to it, i.e., the grade a user provides to an item. There is no such thing as dimension reduction or model fitting. There are mostly two sections:

*Viewer filtering*: In this case, if user A's qualities are comparable to those of another user B, then A gets recommended the things B liked. "Users who enjoy products comparable to yours also liked those things", we can make a statement. As a result, we recommend that you use the correlation between two users in this case. Customised recommendation systems might provide viewers with recommendations in the entertainment or video areas. Information about his 'appetite' increases the user's read performance, cuts down on time it takes for them to choose content, and can even pique their interest in the product. Kumar et al. (2019a). Recommended websites can now obtain user behaviours such as length of stay, favourite links, and the number of likes. The two types of review activities are obvious and implicit feedback behaviours. Explicit feedback actions can reveal user preferences directly (Sunitha and Adilakshmi, 2018). The user's selections cannot be clarified through different prediction conduct. Chang et al. (2016). The user's web browsing history is a typical form of implicit feedback. When viewing the web, the user may not be interested in the item but may click to view it out of curiosity or unwittingly (Cai et al., 2020).

Even though the record does not clearly understand the user's preferences, the implicit data gathered by the general website accounts for a significant amount. As a result, it is important to dig out the different prediction data values meaning content, depending on the use of explicit data, to create personalised suggestions. The algorithms required for personalised recommendation are divided into two types, depending on the data sources: the first is knowing how well specific content is, and the second is a know-how good based on collaborative filtering. The main idea of the entertainment recommender is to endorse to users the details that have the highest similarity in content between the items they like and the data they have followed; the recommender method essentially uses a cohesive median filter, and its core advice idea is as follows: the user has many other users with similar preferences, and then recommends taking to the user items that other users have purchased; however, this use of the personalised recommender systems is limited.

The remainder of the paper is organised as follows: Section 2 presents the literature review about existing techniques of the online web –recommendation system, Section 3 presents a proposed solution to improve the performance of content-based filtering, Section 4 presents a comparative analysis using the proposed approach, and finally, Section 5 presents the concluding remarks about proposed work.

## 2 Literature survey

This section presents the description of existing online web recommendation systems techniques, which mainly includes discussing collaborative and content-based filtering. Nilashi et al. (2018) developed a hybrid approach using collaborative filtering (CF). Mainly this technique focused on sparsity and scalability issues by incorporating dimensionality reduction and ontology schemes. Further, this scheme uses the singular value decomposition (SVD) technique to improve scalability. Wei et al. (2017) discussed that CF schemes suffer from a cold start problem where no rating is available and an incomplete cold start problem where very few ratings are available. Due to these scenarios, the CF-based recommender system fails to provide an accurate recommendation. Wei et al. (2017) developed the CF approach based on the deep learning neural network to overcome these issues. Wang et al. (2018) developed a content-based filtering method for publication recommendations. This scheme follows a hybrid approach, combining chi-square feature selection and softmax regression modules to improve the recommendation performance.

Chae et al. (2018) used generative adversarial networks (GAN) for collaborative filtering to improve recommendation accuracy. Unlike conventional methods, the authors introduced a vector-wise adversarial training approach for the CF model. Son et al. (2017) reported that content-based filtering (CBF) schemes use the correlation between contents to improve the recommendation. The correlations are obtained using item information, which helps compute the similarity. The authors developed a CBF scheme that uses a multi-attribute network to recommend items to users in this work. This network considers the linked items and measures the direct and indirect similarity.

Further, centrality and clustering mechanisms are also considered in this model, which identifies mutual relationships and patterns among the items. Alhijawi and Kilani (2016) focused on heuristic approaches for CF and developed a genetic algorithm to measure the similarity between users and items. This scheme, SimGen, uses Pearson correlation and vector cosine-based similarity measurements to establish the item-user interaction for the recommendation. Like this approach, Logesh et al. (2018) also focused on bio-inspired algorithms for CF-based recommender systems. This scheme uses a combined model of CF and clustering to group the users. However, the traditional clustering algorithms do not achieve the desired performance for recommender systems.

Hence, a new bio-inspired clustering algorithm uses swarm intelligence and a fuzzy clustering model for collaborative filtering. Ali et al. (2018) developed a content-based filtering method for movie recommendation with the help of genomic tags of movies. The unnecessary tags cause the recommendation difficulty, and principal component analysis (PCA) and Pearson correlation techniques are applied to minimise the tags and reduce the complexity. Lian et al. (2017) focused on overcoming the data sparsity problem by developing a cross-domain recommendation system named CCCFNet. However, this is a combined approach of collaborative and content-based filtering. Later, a MAP model for embedding into a multi-view neural network. Chen et al. (2019) developed joint neural collaborative filtering (J-NCF) model for a movie recommendation system. This approach is based on the deep feature learning and deep interaction model with a rating matrix. The deep feature model generates the user-item rating matrix, and the deep interaction model captures non-linear user-item interactions. Thus, this approach considers point-wise and pair-wise loss puncturing the training process to improve the recommendation accuracy.

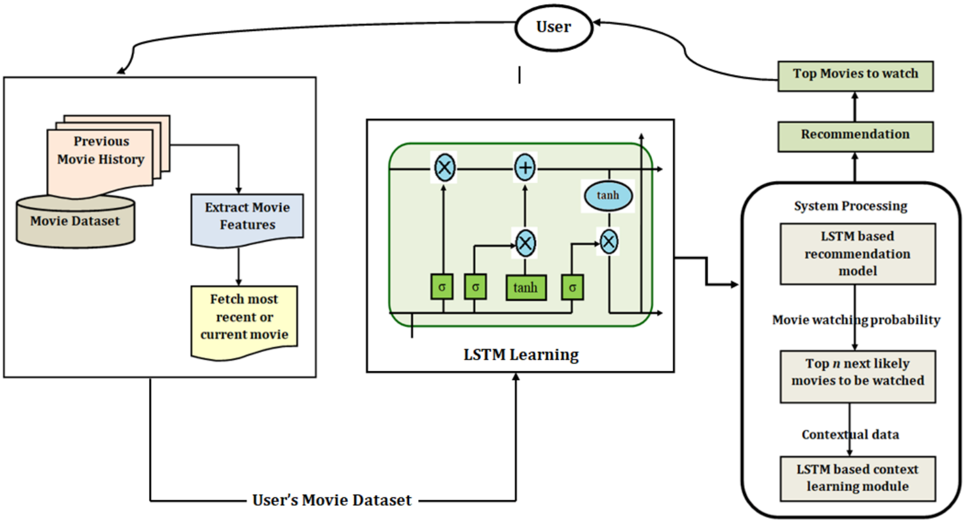
Liu et al. (2018) reported that conventional CF methods suffer from limited learning capacity, data sparsity, and cold-start problems. Thus, a deep learning-based model is introduced, which uses autoencoders. This scheme uses a combination of content-based filtering and collaborative filtering with autoencoders. Mohammadpour et al. (2019) developed a single-objective hybrid evolutionary computation approach for offline collaborative filtering RS. The performance of the genetic algorithm is further improved by incorporating gravitational emulation local search (GELS) algorithm. The dimensionality reduction can be made by several methods like SVD, probability matrix factorisation, and non-negative matrix factorisation.

## 3 Proposed model

The content-based filtering-based schemes are widely adopted for various recommender systems. In this approach, we use learning-based methods to improve the performance of the recommendation system. Due to their efficient learning, deep learning-based schemes are widely adopted in various machine learning applications. We used Python libraries like a Numpy for mathematical calculations, Pandas for Dataframe manipulations, Seaborn and matplotlib for detailed visualisations, IPE widgets for interactive analysis, Jupiter notebook for interactive shells, and at last is, set the figure size and background for our visualisations. In this case, set the figure size as 16 units, four *x*-axes, and five units for the *y*-axis similarly, we can also select the five thirty-eight backgrounds for our plots and another background via charts.

In this work, we develop a deep learning-based approach for a movie recommendation system based on ratings and interactions with other users. Figure 2 shows the proposed architecture of the deep learning-based model for the recommender system.

**Figure 2** The overall architecture of the recommender systems (see online version for colours)



Deep neural network schemes are widely adopted in natural language processing (NLP) systems (Figure 3). In this field of deep learning, recurrent neural networks have gained attraction for sequence prediction problems. Specifically, the LSTM (Long-Short Term Memory) is used as a recurrent neural network system. In this work, we present a learning-based recommender model using content-based filtering.

## *3.1 LSTM*

This section briefly describes the LSTM model for pattern learning based on deep learning. Generally, LSTM is a type of RNN used widely in various pattern learning schemes of data mining. The LSTM model consists of multiplicative gates, which help control the flow of information from input to output or vice-versa to various network states. The hidden layer of LSTM receives the input from the input layer and the previous hidden state as $x_t$ and $h_{t-1}$, respectively. The input can be expressed as:

$$\hat{h}_t = W_{hx} x_t + W_{hh} h_{t-1} \quad (1)$$

The LSTM network has three gating units as input gate, output gate, and forgets gate denoted as $i$, $o$, and $f$, respectively. These stages are represented as:

$$i_t = \sigma\left(W_{ix} x_t + W_{ih} h_{t-1}\right)$$

$$o_t = \sigma\left(W_{ox}x_t + W_{oh}h_{t-1}\right)$$

$$f_t = \sigma\left(W_{fx}x_t + W_{fh}h_{t-1}\right) \tag{2}$$

where $\sigma$ denotes the logistic function. At this phase, the input gate manages the input level to each hidden unit to be written to the internal state vector $c_t$, forget gate controls the amount of information preserved in the previous internal state $c_{t-1}$. The forget gate is used to analyse the information which can be discarded in the current phase. Further, the combination of writing and forget gate is used to control the information which needs to be stored or overwritten for each time-stamp. The internal state can be presented by

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh\left(\hat{h}_t\right) \tag{3}$$

Moreover, the output gate helps identify how much information each unit's activation function should be preserved. This helps preserve the information that is not useful for this state but may be used in upcoming states. The final output of the hidden layer can be expressed as:

$$h_t = \tanh\left(c_t\right) \odot o_t \tag{4}$$

## *3.2 Multiplicative LSTM*

The traditional LSTM learning model is further improved, and a multiplicative LSTM (mLSTM) has been developed recently. The multiplicative LSTM considers it factorised hidden-to-hidden transitions of mRNN and combines them with the gating framework from LSTM. The traditional mRNN and LSTM models can be combined by adding connections from the intermediate state of mRNN $m_t$ to each gating unit. The updated gate configuration can be presented as:

$$m_t = \left(W_{mx}x_t\right) \odot \left(W_{mh}h_{t-1}\right)$$

$$\hat{h}_t = W_{hx}x_t + W_{hm}m_t$$

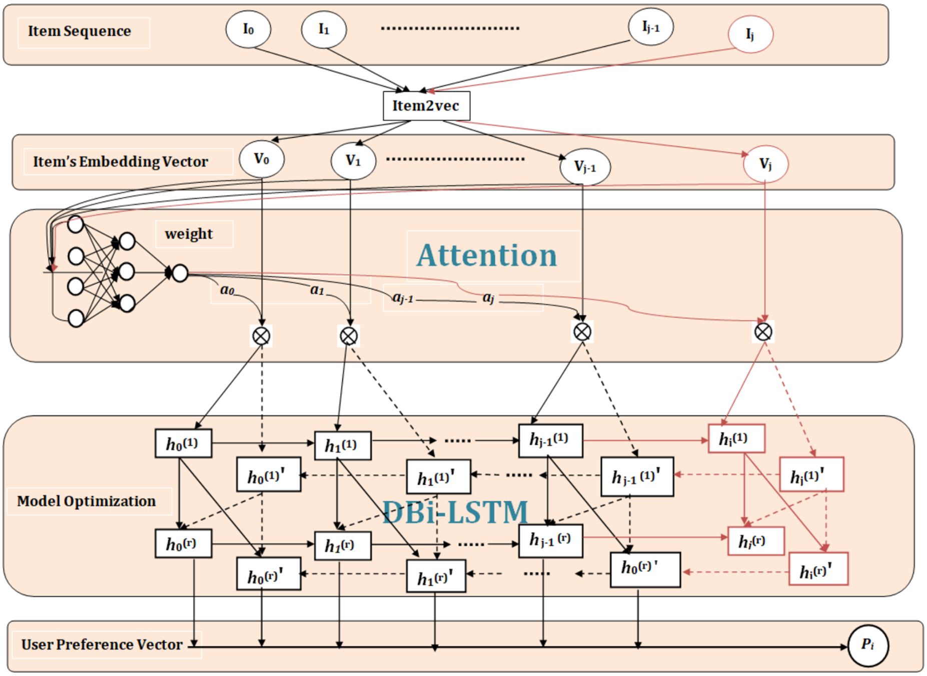
$$i_t = \sigma\left(W_{ix}x_t + W_{im}m_t\right)$$

$$o_t = \sigma\left(W_{ox}x_t + W_{om}m_t\right)$$

$$f_t = \sigma\left(W_{fx}x_t + W_{fm}m_t\right) \tag{5}$$

Mainly, this architecture combines the input-dependent transitions of mRNN with a long time lag and information control of LSTMs. The gated units of LSTM help control and bypass the outcome of the hidden weight matrix. Moreover, the sigmoid input and forget gates help to control the more complex transitions compared with the traditional mRNNs.

**Figure 3** Proposed deep learning-based recommender system (see online version for colours)



## 3.3 Proposed solution

This section presents the proposed solution for the movie recommendation system by using LSTM and deep learning architecture. The proposed approach is divided into three phases: item embedding process, weight updating process, model optimisation, and user preference learning.

## 3.4 Embedding process

Let us consider that $S$ denotes the user interaction item set as $S=\left(S_1,S_2,S_3,\ldots,S_m\right)$ where $S_i$ is the item sequence of *i*th user is given as $S_i=\left\{I_1,I_2,I_3,\ldots,I_n\right\}$. First, we focus on generating the low-dimensional feature vectors from the item set. We select the sequences that show feedback on user preferences in this work. The conventional approach considers only a second-order correlation between items. In this work, we incorporate class labels of items into item vectors, which helps improve the relevancy between item and user preferences. This approach treats each item as a word vector embedded into a vector of fixed dimensions. We formulate an optimisation problem where our goal is to maximise the following objective function:

$$\frac{1}{M}\sum_{x=1}^{M}\sum_{y\neq i}^{M}\log p\left(I_y \mid I_x\right) \tag{6}$$

where $M$ denotes the length of the item sequence of interactions, $p\left(I_y \mid I_x\right)$ denotes the SoftMax function as follows:

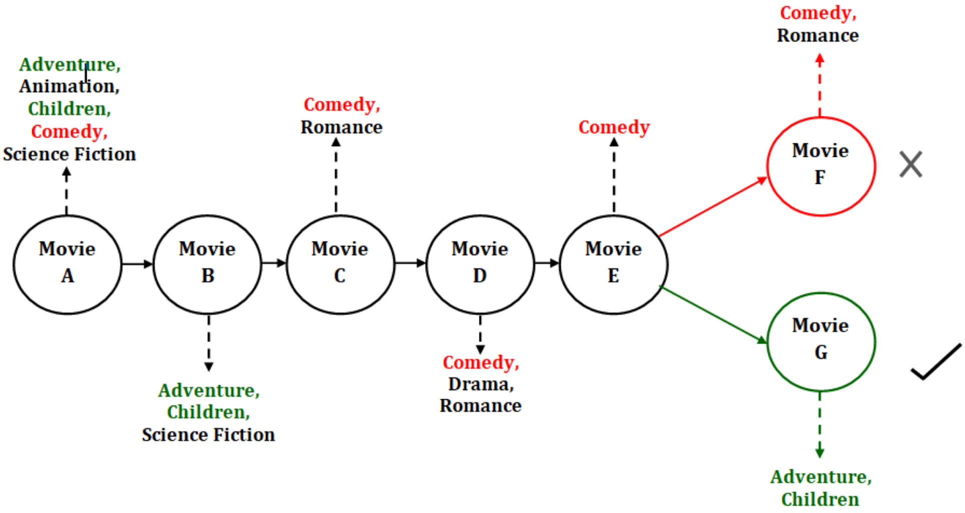
$$p\left(I_y \mid I_x\right)=\sigma\left(I_x^T I_y\right)\prod_k^N \sigma\left(-I_x^T I_j\right) \tag{7}$$

where $\sigma(*)$ is the sigmoid activation function, and $N$ denotes the number of negative samples in the positive sample set. With the help of the embedding process, the item sequence can be represented as $S_i=\{V_1,V_2,V_3,\ldots,V_n\}$

## *3.5 Weight updating process*

Currently, attention mechanisms are widely adopted as a promising solution for sequential data processing, such as speech recognition, machine translation, part-of-speech tagging, and many more. The attention approach uses a weighted transformation model to establish the correlation between highlighted key data. This highlights the important local information. Let us consider an example where a user interacts with different movies such as ‘A’, ‘B’, ‘C’, ‘D’, and ‘E’. Here, we aim to predict and suggest similar types of movies. Figure 4 illustrates the process of predicting the next movie.

**Figure 4** User’s interest-based movie recommendation (see online version for colours)



According to this scenario, the ‘F’ movie is recommended based on the most recent interaction of users with movies C, D, and E. In contrast, complete interaction analysis shows that the ‘G’ movie can be considered the updated recommendation based on the previous interactions. Thus, we present a self-attention mechanism to realise the relation between user interactions. The weight updating process uses the self-attention mechanism. Initially, this scheme traverses each encoder state to obtain the relation between the target and the source item. Based on this relation, it generates a score for each state in the encoder.

Further, this approach uses a SoftMax function to normalise the scores. These normalised scores generate the probability distribution of a given target item state. These two steps help obtain item weight information from the given distribution. Here, the item features are mapped from a given dimensional space to another. The relationship between these mappings can be expressed as follows:

$$I_z = f_{Relu}\left(WI_d + b\right)$$

$$A = Softmax\left(I_z W\left(I_d\right)^T\right)$$

$$I_z = AI_z \tag{8}$$

where $W$ denotes the weight matrix, $b$ represents the bias and $I_z$ is the feature representation vector obtained from the feature embedding process. The first part of equation (8) denotes that the d-dimensional feature of user interaction is mapped to z-dimensional space. The second part shows the weight computation for all user interactions. The weight matrix is processed using a loss function which helps to adjust the weights automatically during the training phase. Similarly, the third part of the equation shows the final output of the self-attention mechanism.

## *3.6 Learning module*

In this process, we treat user-item interaction as a sentence, and each item is treated as a word. The proposed scheme is based on the deep bidirectional multiplicative LSTM, as depicted in Figure 2. This architecture helps to utilise forward and backward contexts representation. Moreover, this network extracts the characteristics efficiently. This learning modelling architecture consists of a double hidden layer. The lower layer provides the structure of information about the upper layer. During the learning process, the network generated a specific set of parameters and fed it into inter-neurons in the same layer at the later time-stamp *t*. The input sequence of each hidden layer traverse from left to right and right to left. At the time-stamp *t*, the output layer of BiLSTM for $r-1$ layer is considered as an input to the intermediate neuron of *r*th layer.

The movement in the left and right directions is expressed as follows:

$$\vec{h}_t^{(r)} = f\left(\vec{A}^{(r)}\vec{h}_t^{(r-1)} + \vec{B}^{(r)}\vec{h}_{t-1}^{(r)} + \vec{z}^{(r)}\right)$$

$$\overleftarrow{h}_t^{(r)} = f\left(\overleftarrow{A}^{(r)}\overleftarrow{h}_t^{(r-1)} + \overleftarrow{B}^{(r)}\overleftarrow{h}_{t-1}^{(r)} + \overleftarrow{z}^{(r)}\right) \tag{9}$$

For each step of the learning process, the hidden layer produces training results that are further propagated with the help of input parameters. The final output produced by the last hidden layer is expressed as:

$$\vec{P} = concat\left(\vec{h}_t^{(r)}, \overleftarrow{h}_t^{(r)}\right) \tag{10}$$

Here, $\vec{A}$, $\vec{B}$ and $\vec{z}$ denotes the weight matrix and offset vector, which is generated in forwarding propagation at $r$ layer of the model, similarly, $\vec{A}$, $\vec{B}$ and $\vec{z}$ represents the weight matrix and offset vector in backward propagation, $\vec{P}$ is the output vector, $\vec{h}_t^{(r)}$, and $\overleftarrow{h}_t^{(r)}$ denotes the intermediate output vectors.

## 4 Results and discussion

In this section, we present the experimental analysis of the proposed model of the movie recommendation system. The proposed model is implemented on an Intel i5 processor CPU @3.10 GHz, 8GB RAM computer. These experiments are carried out using Python 3 with Jupiter Notebook. We use the MovieLens dataset for this experiment, which is publicly available online. This dataset contains 20 million ratings, and 465,000 tags are applied to a total of 27,000 movies, including 19 genres by 138,000 users. The user rating data is stored in a rating file sorted according to user ID movie ID, and Table 1 shows the sample of item sequence stored in the movie rating file.

**Table 1** Sample of item sequence lying in of movie rating

| *User ID* | *Movie ID* | *Rating* | *Timestamp* |
|---|---|---|---|
| 1 | 122 | 5 | 838985046 |
| 1 | 185 | 5 | 838983525 |
| 1 | 231 | 5 | 838983392 |
| 1 | 292 | 5 | 838983421 |
| 1 | 316 | 5 | 838983392 |

Similarly, Table 2 shows the item's content samples of the considered dataset.

**Table 2** Sample data of item's content for proposed recommendation system

| *User ID* | *Movie ID* | *Genres* |
|---|---|---|
| 1 | Jumanji (1995) | Adventure \| Children \| Fantasy |
| 2 | Grumpier Old Men (1995) | Comedy \| Romance |
| 3 | Waiting to Exhale (1995) | Comedy \| Drama\| Romance |
| 4 | Father of the Bride Part II (1995) | Comedy |
| 5 | Heat (1995) | Action \| Crime \| Thriller |

To measure the performance of the proposed recommendation system, we compute mean absolute error (MAE), a widely used parameter to measure the performance of recommender systems. The MAE is a statistical accuracy measurement obtained by finding the absolute difference between actual and predicted recommendation values. The lower MAE value represents the better prediction. The MAE can be computed as:

$$MAE = \frac{\sum P_{ij} - r_{ij}}{M} \tag{11}$$

where $M$ is the total number of predictions, $P_{ij}$ denotes the predicted value of user $i$ for item $j$ and $r_{ij}$ denotes the actual ratings. Similarly, we compute precision, recall, and F-measure based on the actual and predicted ratings of the movie data. Precision is known as the exactness of measurement. This can be computed as:

$$Precision = \frac{TP}{TP + FP} \tag{12}$$

The recall is the measurement of completeness which determines the number of relevant retrieved items out of all relevant items. This can be expressed as:

$$Recall = \frac{TP}{TP + FN} \tag{13}$$

Similarly, the F-measure is the harmonic mean of precision and recall. This can be computed as follows:

$$F = 2 \times \frac{(precision \times recall)}{(precision + recall)} \tag{14}$$

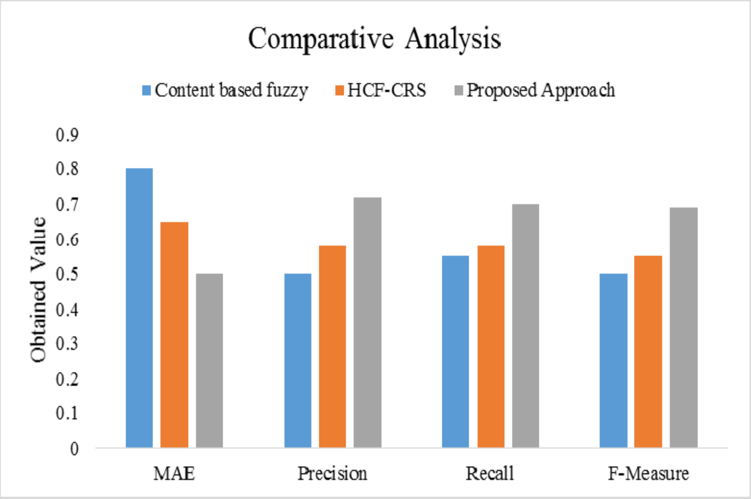
Based on these parameters, we obtained the performance of the recommendation system in terms of MAE, Precision, Recall, and F-Measure. The obtained performance is compared with existing techniques such as content-based fuzzy (Ayyaz et al., 2018) and HCF-CRS (Ayyaz et al., 2018) (Table 3).

**Table 3** Comparative analysis between CBS, HCF-CRS, and proposed approach

| | *MAE* | *Precision* | *Recall* | *F-Measure* |
|---|---|---|---|---|
| Content based fuzzy | 0.8 | 0.5 | 0.55 | 0.5 |
| HCF-CRS | 0.65 | 0.58 | 0.58 | 0.55 |
| Proposed approach | 0.5 | 0.72 | 0.7 | 0.69 |

This comparative study is illustrated in Figure 5, which shows the robustness of the proposed approach.

**Figure 5** Comparative analysis (see online version for colours)



Furthermore, we compare the precision performance for different numbers of predictions. Table 4 shows the comparative analysis in terms of precision and MAP. The obtained performance is compared with the existing techniques (Zhao et al., 2020).

**Table 4** Movie recommendation performance for varied level of retrieval

| *Technique* | *Precision @3* | *Precision @5* | *Precision @10* | *MAP* |
|---|---|---|---|---|
| BPR | 0.2795 | 0.2664 | 0.2301 | 0.3549 |
| PRFM | 0.2884 | 0.2699 | 0.2481 | 0.3885 |
| LambdaFM | 0.3108 | 0.2953 | 0.2612 | 0.4014 |
| RRN | 0.2893 | 0.274 | 0.248 | 0.3631 |
| IRGAN | 0.3022 | 0.2885 | 0.2582 | 0.3744 |
| CTR | 0.2824 | 0.2694 | 0.2493 | 0.3725 |
| CDL | 0.2875 | 0.2731 | 0.2504 | 0.3863 |
| ConvMF | 0.2901 | 0.2856 | 0.2545 | 0.3996 |
| LSIC-V1 | 0.2946 | 0.2713 | 0.2531 | 0.4066 |
| LSIC-V2 | 0.3004 | 0.2843 | 0.2567 | 0.4101 |
| LSIC-V3 | 0.3105 | 0.3023 | 0.261 | 0.4163 |
| LSIC-V4 | 0.3327 | 0.3173 | 0.2847 | 0.4321 |
| Proposed approach | 0.3511 | 0.3385 | 0.3218 | 0.4951 |

These comparative studies show that the proposed approach achieves improved precision, recall, f-measure, and MAP for varied experiments.

## 5 Conclusion

The major objective of this work is to provide a new method for improving the performance of a movie recommendation system. Content-based filtering algorithms are currently frequently employed in a variety of recommender systems. However, there are several difficulties with these strategies. We propose a deep learning-based recommender system model to address these issues. The proposed model employs deep learning and a multiplicative BILSTM model to improve learning. This method starts with the generation of an item sequence and the formulation of a vector. A weight updating technique is used to improve the suggestion for these vectors. Finally, multiplicative BILSTM is used to add a learning process. The proposed strategy yields superior suggestion accuracy when compared to state-of-the-art methodologies, according to the comparison analysis.

## References

Achakulvisut, T., Acuna, D.E., Ruangrong, T. and Kording, K. (2016) ‘Science concierge: a fast content-based recommendation system for scientific publications’, *PloS One*, Vol. 11, No. 7, e0158423.

Alhijawi, B. and Kilani, Y. (2016) ‘Using genetic algorithms for measuring the similarity values between users in collaborative filtering recommender systems’, *2016 IEEE/ACIS 15thInternational Conference on Computer and Information Science (ICIS)*, IEEE, Okayama, Japan, pp.1–6, doi: 10.1109/ICIS.2016.7550751.

Ali, S.M., Nayak, G.K., Lenka, R.K. and Barik, R.K. (2018) 'Movie recommendation system using genome tags and content-based filtering', *Advances in Data and Information Sciences*, Springer, Singapore, pp.85–94.

Ayyaz, S., Qamar, U. and Nawaz, R. (2018) 'HCF-CRS: a hybrid content based fuzzy conformal recommender system for providing recommendations with confidence', *PloS One*, Vol. 13, No. 10, e0204849.

Cai, X., Hu, Z., Zhao, P., Zhang, W. and Chen, J. (2020) 'A hybrid recommendation system with many-objective evolutionary algorithm', *Expert Systems with Applications*, Vol. 159, 113648.

Chae, D.K., Kang, J.S., Kim, S.W. and Lee, J.T. (2018) 'Cfgan: a generic collaborative filtering framework based on generative adversarial networks', *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, Fully online, pp.137–146, https://doi.org/10.1145/3269206.3271743

Chang, P.C., Lin, C.H. and Chen, M.H. (2016) 'A hybrid course recommendation system by integrating collaborative filtering and artificial immune systems', *Algorithms*, Vol. 9, No. 3, p.47.

Chauhan, D., Kumar, A., Bedi, P., Athavale, V.A., Veeraiah, D. and Pratap, B.R. (2021) 'An effective face recognition system based on cloud based ioT with a deep learning model', *Microprocessors and Microsystems*, Vol. 81, 103726, https://doi.org/10.1016/j.micpro.2020.103726

Chen, W., Cai, F., Chen, H. and Rijke, M.D. (2019) 'Joint neural collaborative filtering for recommender systems', *ACM Transactions on Information Systems* (*TOIS)*, Vol. 37, No. 4, pp.1–30.

Chu, W.T. and Tsai, Y.L. (2017) 'A hybrid recommendation system considering visual information for predicting favorite restaurants', *World Wide Web*, Vol. 20, No. 6, pp.1313–1331.

Deldjoo, Y., Elahi, M., Cremonesi, P., Garzotto, F., Piazzolla, P. and Quadrana, M. (2016) 'Content-based video recommendation system based on stylistic visual features', *Journal on Data Semantics*, Vol. 5, No. 2, pp.99–113.

Karn, A.L., Pandya, S., Mehbodniya, A., Arslan, F., Sharma, D.K., Phasinam, K., Aftab, M.N., Rajan, R., Bommisetti, R.K. and Sengan, S. (2021) 'An integrated approach for sustainable development of wastewater treatment and management system using IoT in smart cities', *Soft Computing*, pp.1–17.

Katarya, R. and Verma, O.P. (2017) 'An effective collaborative movie recommender system with cuckoo search', *Egyptian Informatics Journal*, Vol. 18, No. 2, pp.105–112.

Kumar, D., Kumar, S. and Bansal, R. (2020) 'Multi-objective multi-join query optimisation using modified grey wolf optimisation', *International Journal of Advanced Intelligence Paradigms*, Vol. 17, Nos. 1–2, pp.67–79.

Kumar, D., Kumar, S., Bansal, R. and Singla, P. (2017) 'A survey to nature inspired soft computing', *International Journal of Information System Modeling and Design (IJISMD)*, Vol. 8, No. 2, pp.112–133.

Kumar, D., Mehrotra, D. and Bansal, R. (2018) 'Metaheuristic policies for discovery task programming matters in cloud computing', *2018 4th International Conference on Computing Communication and Automation (ICCCA)*, Greater Noida, India, pp.1–5, http://doi.org/10.1109/Ccaa.2018.8777579

Kumar, P., Kumar, V. and Thakur, R.S. (2019a) 'A new approach for rating prediction system using collaborative filtering', *Iran Journal of Computer Science*, Vol. 2, No. 2, pp.81–87.

Kumar, D., Mehrotra, D. and Bansal, R. (2019b) 'Query optimization in crowd-sourcing using multi-objective ant lion optimizer', *International Journal of Information Technology and Web Engineering (IJITWE)*, Vol. 14, No. 4, pp.50–63.

Lazar, A.J.P., Sengan, S., Cavaliere, L.P.L., Nadesan, T., Sharma, D., Gupta, M.K., Palaniswamy, T., Vellingiri, M., Sharma, D.K. and Subramani, T. (2021) 'Analysing the user actions and location for identifying online scam in internet banking on cloud', *Wireless Personal Communications*, pp.1–24.

Lian, J., Zhang, F., Xie, X. and Sun, G. (2017) 'CCCFNet: a content-boosted collaborative filtering neural network for cross domain recommender systems', *Proceedings of the 26th International Conference on World Wide Web Companion(IW3C2)*, pp.817–818, published under Creative Commons CC BY 4.0 License, 3–7 April, 2017, Perth, Australia, ACM 978-1-4503-4914-7/17/04, http://dx.doi.org/10.1145/3041021.3054207

Liu, Y., Wang, S., Khan, M.S. and He, J. (2018) 'A novel deep hybrid recommender system based on auto-encoder with neural collaborative filtering', *Big Data Mining and Analytics*, Vol. 1, No. 3, pp.211–221.

Logesh, R., Subramaniyaswamy, V., Malathi, D., Sivaramakrishnan, N. and Vijayakumar, V. (2018) 'Enhancing recommendation stability of collaborative filtering recommender system through bio-inspired clustering ensemble method', *Neural Computing and Applications*, pp.1–24.

Low, Y.H., Yap, W.S. and Tee, Y.K. (2018) 'Convolutional neural network-based collaborative filtering for recommendation systems', *International Conference on Robot Intelligence Technology and Applications,* Springer, Singapore, pp.117–131.

Mohammadpour, T., Bidgoli, A.M., Enayatifar, R. and Javadi, H.H.S. (2019) 'Efficient clustering in collaborative filtering recommender system: hybrid method based on genetic algorithm and gravitational emulation local search algorithm', *Genomics*, Vol. 111, No. 6, pp.1902–1912.

Nagpal, S., Athavale, V.A., Saini, A.K. and Sharma, R. (2020) 'Indian health care system is ready to fight against COVID-19 A machine learning tool for forecast the number of beds', *2020 Sixth International Conference on Parallel, Distributed and Grid Computing (PDGC)*, Jaypee University of Information Technology, Waknaghat, Solan, Himachal Pradesh, India, pp.61–65, https://doi.org/10.1109/pdgc50313.2020.9315825

Narayanasami, S., Butta, R., Govindaraj, R., Choudhary, S.S., Sharma, D.K., Poonia, A., Sengan, S., Dadheech, P., Shukla, N.K. and Verma, R. (2021) 'An enhanced trust-based kalman filter route optimization technique for wireless sensor networks', *Wireless Personal Communications*, pp.1–19.

Neffati, O.S., Setiawan, R., Jayanthi, P., Vanithamani, S., Sharma, D., Regin, R., Mani, D. and Sengan, S. (2021) 'An educational tool for enhanced mobile e-learning for technical higher education using mobile devices for augmented reality', *Microprocessors and Microsystems*, Vol. 83, p.104030.

Nilashi, M., Ibrahim, O. and Bagherifard, K. (2018) 'A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques', *Expert Systems with Applications*, Vol. 92, pp.507–520.

Palomares, I., Browne, F. and Davis, P. (2018) 'Multi-view fuzzy information fusion in collaborative filtering recommender systems: application to the urban resilience domain', *Data and Knowledge Engineering*, Vol. 113, pp.64–80.

Paradarami, T.K., Bastian, N.D. and Wightman, J.L. (2017) 'A hybrid recommender system using artificial neural networks', *Expert Systems with Applications*, Vol. 83, pp.300–313.

Raghuwanshi, S.K. and Pateriya, R.K. (2019) 'Collaborative filtering techniques in recommendation systems', *Data, Engineering and Applications*, Springer, Singapore, pp.11–21.

Renuka, S., Kiran, G.R. and Rohit, P. (2021) 'An unsupervised content-based article recommendation system using natural language processing', *Data Intelligence and Cognitive Informatics*, Springer, Singapore, pp.165–180.

Rutkowski, T., Romanowski, J., Woldan, P., Staszewski, P., Nielek, R. and Rutkowski, L. (2018) 'A content-based recommendation system using neuro-fuzzy approach', *2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, IEEE, pp.1–8.

Shu, J., Shen, X., Liu, H., Yi, B. and Zhang, Z. (2018) 'A content-based recommendation algorithm for learning resources', *Multimedia Systems*, Vol. 24, No. 2, pp.163–173.

Son, J. and Kim, S.B. (2017) 'Content-based filtering for recommendation systems using multiattribute networks', *Expert Systems with Applications*, Vol. 89, pp.404–412.

Sunitha, M. and Adilakshmi, T. (2018) 'Music recommendation system with user-based and item-based collaborative filtering technique', *Networking Communication and Data Knowledge Engineering*, Springer, Singapore, pp.267–278.

Wang, D., Liang, Y., Xu, D., Feng, X. and Guan, R. (2018) 'A content-based recommender system for computer science publications', *Knowledge-Based Systems*, Vol. 157, pp.1–9.

Wang, L., Rao, Y., Bian, Q. and Wang, S. (2020) 'Content-based hybrid deep neural network citation recommendation method', *International Conference of Pioneering Computer Scientists, Engineers and Educators*, Springer, Singapore, pp.3–20.

Wei, J., He, J., Chen, K., Zhou, Y. and Tang, Z. (2017) 'Collaborative filtering and deep learning-based recommendation system for cold start items', *Expert Systems with Applications*, Vol. 69, pp.29–39.

Yang, F. (2018) 'A hybrid recommendation algorithm–based intelligent business recommendation system', *Journal of Discrete Mathematical Sciences and Cryptography*, Vol. 21, No. 6, pp.1317–1322.

Zhao, W., Wang, B., Yang, M., Ye, J., Zhao, Z., Chen, X. and Shen, Y. (2020) 'Leveraging long and short-term information in content-aware movie recommendation via adversarial training', *IEEE Transactions on Cybernetics*, Vol. 50, No. 11, November, pp.4680–4693, doi: 10.1109/TCYB.2019.2896766, Epub 2019 Feb 15, PMID: 30794200.