Indonesian Journal of Electrical Engineering and Computer Science
Vol. 36, No. 3, December 2024, pp. 1849~1856
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v36.i3.pp1849-1856

Hesitant fuzzy clustering with convolutional spiking neural network for movie recommendations

Vineet Shrivastava, Suresh Kumar
Department of Computer Science and Engineering, Manav Rachna International Institute of Research and Studies Faridabad, Faridabad, India

Article Info

Article history:

Received Mar 29, 2024
Revised Aug 20, 2024
Accepted Aug 31, 2024

Keywords:

Accuracy
Clustering
Machine learning
Neural network
Recommender system

ABSTRACT

The movie recommender system is one of the most influential and practical tools for aiding individuals in quickly selecting films to watch. Despite numerous academic efforts to employ recommender systems for various purposes, such as movie-watching and book-buying, many studies have overlooked user-specific movie recommendations. This paper introduces a novel approach for movie recommendations that combines the hesitant fuzzy clustering with a convolutional spiking neural network movie recommender system. The initial step involves acquiring input data from benchmark datasets like MovieLens 100K and MovieLens 1M. Further, content-based features are extracted from the dataset using ternary pattern and discrete wavelet transforms. After that hesitant fuzzy linguistic Bi-objective clustering (HFLBC) is applied for cluster selection based on the extracted features. Subsequently, a movie recommender scheme utilizing a convolutional spiking neural network is introduced to predict user film preferences. The efficiency of the proposed model is compared to existing methods such as multi-modal trust-dependent recommender scheme and graph-dependent hybrid recommendation scheme. The results show a significant improvement, with the proposed model achieving 13.79% and 16.47% higher accuracy than the existing methods. The findings highlight the potential of proposed system in enhancing the accuracy and personalization of movie recommendations.

Corresponding Author:

Vineet Shrivastava
Department of Computer Science and Engineering
Manav Rachna International Institute of Research and Studies Faridabad
Faridabad, Haryana, India
Email: vineet.shrivastava62@gmail.com

1. INTRODUCTION

Recommendation systems (RS) serve a purpose similar to that of advisors, specifically designed to manage the rapid expansion of information on the internet [1]. These technologies enable consumers to access personalised goods that will pique their interest. RS have experience working in a variety of fields, including tourism, e-commerce, journalism, and movies [2]. The current deep learning models do not take into consideration the side information about users that is strongly associated to user rating [3]. In fact, combining deep learning with side information may help us produce better answer to difficulties at hand [4].

Traditional systems for recommendation mostly utilize content-based or collaborative filtering techniques, each of which has advantages and disadvantages of its own [5]. Content-based approaches utilize item attributes to promote items similar to those that the user has previously engaged with, whereas collaboration based approach analyzes user-item interactions to identify trends and provide suggestions.

However, both methods have certain disadvantages, such as limited data availability, cold start problem, and the inability to capture intricate customer preferences [6].

Movie recommender systems have advanced from simple methods to complex methodology as technology has advanced. These systems can be categorized into two main groups: collaborative filtering, which is based on user actions and preferences, and content-based filtering, which takes into account the inherent characteristics of movies [7]. However, contemporary movie recommender systems are increasingly adopting hybrid models due to their recognition of the limits of solo solutions. These systems try to address the challenges of cold start problems, data sparsity, and the diversity of user preferences by utilizing a combination of collaborative and content-based methodologies [8].

As movie recommender systems advance with innovation and more personalization, accurately assessing their effectiveness demands a careful grasp of essential factors. Effectiveness is not solely measured by the accuracy of recommendations but also encompasses aspects such as diversity, novelty, and user satisfaction [9]. Therefore, assessing a movie recommender system involves a multifaceted approach considering various dimensions to ensure a holistic evaluation.

This paper presents a novel hybrid approach called the hesitant fuzzy linguistic Bi-objective clustering (HFLBC) with deep convolutional spiking neural network-endorsed movie recommender system (HFLBC-DCSNN-MRS) to address these issues. This framework is compared with two recent hybrid approaches: the multi-modal trust-dependent recommender scheme (MT-ML-MRS) and the graph dependent hybrid recommender scheme (GHRS). It shows significant improvements in terms of accuracy, precision, recall, and F-measure for movie RS.

The structure of the paper is as follows: in section 2, a review of the previous study is given. In section 3 explains and spells out the exact study method. In section 4 is dedicated to the examination of the outcomes and evaluation of the performance, while section 5 provides a concise overview of the paper's discoveries and recommendations for future investigation.

2. LITERATURE SURVEY

This study conducts a comprehensive literature review from 2014 to 2023, utilizing reputable databases like ACM, IEEE, Springer, MDPI, and Scopus. Meticulous selection ensures a thorough examination of the field, capturing the latest developments and enhancing the study's reliability and relevance. Vahedian [10] introduced WHyLDR, a versatile hybrid recommendation model for social media data, offering flexibility and efficiency compared to single-purpose systems, demonstrated in social tagging applications. Zhang *et al.* [11] presented a hybrid recommender system emphasizing user-recommender interaction, which outperformed non-hybrid approaches on the MovieLens dataset. Their study redefined evaluation metrics and discussed various related research areas, enhancing the development of personalized recommender systems. Strub *et al.* [12] introduce a new approach for matrix completion in recommender systems utilizing neural networks and autoencoders, emphasizing the incorporation of side information for cold users/items. Their provision of reusable Lua/Torch code enhances its value for both researchers and practitioners in the field. Xu and Zhang [13] introduce a collaborative filtering recommendation algorithm incorporating hybrid similarity to improve recommendation quality for large data processing, outperforming traditional methods in experimental evaluations. In their proposal for mobile movie RS, Wang *et al.* [14] introduced a sentiment-enhanced hybrid recommender on the Spark platform, aiming to improve accuracy and relevance by integrating collaborative filtering, content-based methods, and sentiment analysis, though lacking comparative analysis. Various approaches, including collaborative filtering, content-based filtering, and hybrid methods, have been developed to enhance user experience by providing personalized recommendations. Notably, progress in deep learning and graph neural networks has made suggestions even more accurate and varied, dealing with problems like lack of data and modelling user preferences [15].

In their work Çano and Morisio [16] provides an extensive literature review on hybrid recommender systems, that covers diffrent aspects about the hybrid recommendations from problems that may occur and the future research directions of the application of this type of recommendation. In context of deep learning techniques Kumar and Bhasker [17] discuss the advantages of them in hybrid recommender systems, elaborating its potential for complex pattern capture and extension to ranking scenarios. To tackle problems like cold start and data sparsity issues. Tahmasebi *et al.* [18] introduce a hybrid recommender incorporating social influence and deep learning that shows high accuracy. A MT-ML-MRS for movies proposed by Choudhury *et al.* [19] that combines user similarity and trust base to address cold start and malicious attacks along with a DNN-based model obtaining high accuracy.

Combining various techniques requires a comparative analysis among them. To show that Shrivastava and Kumar [20] give a comparative analysis of hybrid recommender systems, highlighting their role in enhancing user experience and revenue. In most of the work, autoencoders are used for recommendations. In a work proposed by Darban and Valipour [21] a novel GHRS combining a graph-based model with autoencoder feature extraction addressed the cold-start problem and enhanced recommendation accuracy effectively, mainly in movie recommendation. Visual datasets like posters can also be helpful in terms of recommendation. A hybrid approach provided by Shrivastava and Kumar [22], emphasized the influence of movie posters and advocated for deep learning techniques and hybrid models to overcome existing limitations, integrating deep neural networks with a hybrid combination of content and collaborative filtering enhanced recommendation. The study of multiple research articles on recommender systems highlights significant advancements in mentioning challenges like the cold start problem, data sparsity, and the need of more accurate recommendations. However, a notable gap emerges a lack of a unified, standardized hybrid recommender system capable of seamlessly integrating diverse techniques while considering user sentiment, trust, and social influence. Existing models excel in specific contexts but lack a cohesive solution applicable across varied recommendation scenarios. Thus, the identified problem is the absence of a comprehensive hybrid recommender system that effectively tackles challenges, providing a versatile framework for enhanced accuracy, personalization, and user satisfaction, necessitating an innovative and unified approach in the field.

3. PROPOSED METHOD

Following an extensive review of the literature, the proposed approach is designed to address key unresolved issues in the field of RS. Previous methods, such as the MT-ML-MRS and the GHRS, have managed to handle specific challenges like the cold start problem and data sparsity. However, these approaches often lack a unified framework capable of seamlessly integrating diverse techniques. Our proposed method seeks to improve upon these limitations by hybridizing HFLBC-DCSNN-MRS. This approach aims to enhance recommendation accuracy and user satisfaction by considering implicit ratings, feature selection, and user preference modeling. The proposed framework combines novel experimental approaches with standard methodologies to provide a comprehensive solution for movie RS, with the potential for scalability across other domains. It offers a versatile and cohesive strategy that unifies various techniques, addressing previous shortcomings and setting the stage for more advanced and effective RS. The proposed approach presents a novel method for predicting user movie preferences through four main stages: data collection, feature extraction, clustering of similar movies, and user classification within a content-based recommendation system framework.

3.1. Stages in the proposed approach

In the proposed approach, the framework starts with data acquisition from the MovieLens datasets to gather essential information for recommendation. It then employs content-based feature extraction and cluster selection using HFLBC to categorize data effectively. Finally, a DCSNN predicts user preferences, with performance evaluated to ensure accurate and personalized movie recommendations. Every stage is systematically explained to clarify the functionality of the method. Additionally, Figure 1 visually depicts the step-by-step process, aiding in understanding how each component contributes to predicting user movie preference.

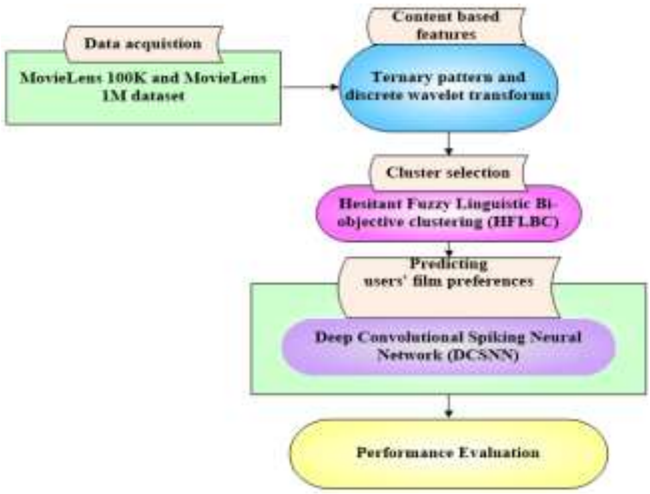

Figure 1. Block diagram of proposed HFLBC-DSCNN-MRS methodology

3.1.1. Data acquisition

This step takes the input data from benchmark datasets such as MovieLens100K [23] and MovieLens1M [24] for the proposed movie recommender system. The MovieLens 100K is a large dataset with 100,000 ratings from 1682 movies and genres. The MovieLens 1M dataset contains about 1-million ratings from 3,883 movies and genres.

RS can use explicit ratings, such as like/dislike or star ratings, but these often require user feedback, which may not always be available or desirable. As a result, many systems generate implicit ratings based on user behavior, such as viewing durations [25]. Since users typically do not explicitly rate films in the system, implicit ratings are calculated using in the (1).

$$r(u,i) = \frac{t(u,i)}{t_i} \tag{1}$$

3.1.2. Content features

Incorporating genre characteristics, ratings, and specific attributes of a movie directly into analysis involves extracting content elements from provided movie summaries [26]. Utilizing word roots identified within these summaries ensures accurate portrayal, with techniques like ternary patterns, discrete wavelet transforms, and with term frequency-inverse document frequency (TF-IDF) calculations recommended for systematic feature gathering. In (2) precisely represents the TF-IDF rating of a term within a film.

The genre characteristics, ratings, and movie were all directly used. The content aspects of a movie were obtained from the movie summary materials. Instead, these terms may be used to indicate content features just as they are now. It is necessary to use the word roots you find in the summary page to portray a film. Gather the film features, use the ternary pattern and discrete wavelet transforms (TF-IDF). The TF-IDF rating (term) j at film i can be represented by (2), where time frequency denotes times count title occurs at film, inverse document frequency denotes. Represents total films count and signifies count of movies have rating. Examples of content characteristics for two distinct films are shown in Table 1.

$$tf - idf(j,i) = tf(j,i) \times log(M/m_i) \tag{2}$$

Table 1. Examples of films content characteristics

Movie	Genre	Rating
Toy story	Animation	3
Jumanji	Adventure	4

3.1.3. Content feature-based user profiles

Utilising features like movie, genre, and rating, film recommendation scheme seeks out the media that users could truly desire to watch [27]. Use the movies to assign weight to each user and feature the user has already seen and their implicit evaluations of those films. Use the feature set, including movies, ratings, and genre, in this case. There are 36 categories in training material. Each feature's weight is determined by the implicit user ratings for all training data contents that contain that feature. Feature weight i in feature k for user s can be represented in (3), where P_u^{train} represents movies watched through user during training time. k denotes feature set, $k \in \{genre, movie, rating.$ $t(s,i) \in T$ indicates user implicit rating s for film i and $y_{k,s}(i,j) \in \{0,1\}$ signifies j^{th} film feature i. Film i has feature j, $y_{k,s}(i,j)$ is 1.

$$w_k(s.i) = \frac{1}{P_u^{train}} \times \sum_{j\in} P_u^{train} y_{k,s}(i,j) \times t(s,i) \tag{3}$$

3.1.4. Feature selection utilizing HFLBC

The selection of clusters with the help of HFLBC based on extracted features. HFLBC is a commonly used method for identifying key features in categorization tasks. It relies on implicit ratings provided by viewers for the films they watch. This approach eliminates the need for explicit positive or negative reviews for every film, providing more detailed insights into viewer preferences [28]. As a result, it is not possible to assess the importance of each characteristic using correlation measure, like common info. A characteristic may be essential for one user but not for the others, hence it is necessary to take each user into account independently when calculating the redundancy of the extracted features. In user profile entry with content-based features, $w_k(s,i)$ is corresponds to how many films includes feature i user s has seen and how much enjoyed them. A measure of feature relevance i for s, utilize content feature and is represented in (4).

$$Re\,l\,(s,i) = w_k(s,i) \quad (4)$$

Since there are clusters when two features are correlated, none can be left out. Define correlation among features and j as close to $w_k(s,i)$ values. Subscript-asymmetric linguistic term set (LTS) is displayed on, $R = \{r_\alpha r_0, r_1, \ldots r_{2\beta}$ where r_α denotes the selection of clusters. Here, $r_\alpha$and $r_{2\beta}$ signify lower bound, upper bound of r_α. A uniformly distributed LTS is expressed in (5).

$$R = \{r_0 : selection\ of\ clusters\} \quad (5)$$

For a set of objects, $Y = \{y_j | j = 1,2, \ldots .. m\}$, a hesitant fuzzy linguistic preference relation (HFLPR) is represented as $C = (c_x) \cup Y \times Y$, where c_x indicates that the object y_j is referred to y_i with respect to the hesitancy degrees. $c_y = \{c_y^0 | h = 1,2, \ldots . c_y$ denotes HFLTS, c_y indicates certain terms r_α for all $j, i = extracted\,features\,from\,movies$, and the genre with name is represented in (6), where, c_{ji} denotes actor name. It is annoying if an HFLPR contains a different number of linguistic words, necessitating the addition of some acceptable components to a particular HFLTS to continue the computation.

$$c_i \otimes c_i = r_{2\alpha} \times r_0, c_{ji} = r_\alpha, \# c_{ji} = \# c_{ji} \quad (6)$$

The selection of features was given in (7),

$$selection\ of clusters = \sqrt{\frac{1}{Q} \sum_{q=1}^{Q} \left(\frac{|\alpha_q^1 - \alpha_q^2|}{2\alpha} \right)} \quad (7)$$

where, α_q^1 and α_q^2 are the corresponding linguistic terms and some extracted features and is represented in (8).

$$F(\alpha_q^1, \alpha_q^2) = -\sum_{i=1}^{m} (selection\ of\ clusters) c_{ji} (log\, c_{ji}) \quad (8)$$

Therefore, when selecting the clusters for film recommendation, do not remove redundant features and ignore unnecessary ones. Along with the relevant features, order the relevant attributes and choose just the clusters with highest $w_k(s,i)$ values from (4). The performance of the recommender can be enhanced by feature selection. Finally, the selection of clusters based on extracted features is done through using HFLBC and these features are presented towards the classification phase.

3.1.5. Classification using DCSNN

The DCSNN classification technique is proposed here for movie recommendation [29]. The spiking classifiers consist of two interconnected layers and two spike neural levels that transform spike feature sequences to the target area, determining the classification outcome through time rates and SoftMax. The pooling layer, with a size of two, necessitates the network to operate for a specific duration. Consequently, the classification relies on the mean spike firing rates after a designated time, integrating temporal dynamics for accurate outcome determination.

Mathematically, leaky integrate-and-fire neuron (LIF neuron) can be modelled and is represented in (9), where $\alpha_n = 10$ time constant, $Y(u)$ is membrane layer, V_{reset} is reset layer, $Y(u)$ external stimulus.

$$\alpha_n = \frac{eV(u)}{eu} = -(V(u) - V_{reset}) + Y(u) \quad (9)$$

The regular divergent is approximated using an isolated contrast equation and is represented in (10).

$$\alpha_n(eV(u) - V[u-1]) = -V[u-1] - V_{reset} + Y(u) \quad (10)$$

Spiking neural network that maximizes performance through data-driven updates, particularly when event-driven sensor inputs are merged. In (11) represents the transfer of the original video pixels to the ratings time of neural spikes using the sigmoid function. where $q$are movie pixel at interval, s depicts spike's rating time, α is non-linear coding parameters, S_{max} is extreme time, with group value.

$$s = \frac{S_{max}}{1+exp(\alpha(128-q))} \quad (11)$$

Therefore, instantaneous preference at time t can be represented in (12) and (13).

$$V(t) = \sum_{k=1}^{N} u_k s_k(t) \tag{12}$$

$$V[u] = e(V[u-1], Y(u) = V[u-1] + \frac{1}{\alpha_n} - (V[u-1] - V_{reset}) + Y(u) \tag{13}$$

The following sections represent the membrane layer after a neuron has been charged but before the spike goes away, $V[u]$ represent the membrane layer shortly after the neuron has fired the spike, and represent the spike emitted by the neurons. The unsupervised SNN is trained using an arbitrary technique, as shown in (14).

$$Q_{sp} = 0.5 \sum_j (s_j(u) - s_j(u) \tag{14}$$

Features are supplied as inputs to the bottom layer of DCSNN's hierarchical structure. Convolutional operations have a level of invariance due to the complicated CNN architecture. Then, separate equations of charge and discharge phases represented in (15) and (16),

$$[u] - e(U[u-1], Y[u]) \tag{15}$$

$$H(u) = h(I[u]) - U_{threshold} \tag{16}$$

where $h(I[u])$ is the list of movies. The list of movies with ratings can be represented in (17).

$$h(I[u]) = \begin{Bmatrix} 1x \leq 0 \\ 0x \geq 0 \end{Bmatrix} \tag{17}$$

A spiking neuron generates binary data, and only one run's classified choice may be easily modified. As a result, it is commonly assumed that the result is conveyed by spike firing rate of outcome layer in a period and the resultant movie preference can be represented in (18).

$$V[u] = h[u].(1 - s(u)] + s(u).V(MOVIE\ PREFERENCE) \tag{18}$$

Calculate the preference vector by identifying user interests and considering the volume of user activity. The chance that a user would like to choose an interest in engaging in new activities is displayed for each member of the preference vector. In (19) is used to construct a preference vector for a user and is represented by (19).

$$pref(feature_j) = \sum_{i=1}^{|cluster_j|} user\ movie\ preference(activity) \tag{19}$$

Even with a steady input, the member layer is frequently changed. When the membrane voltage of the neuron reaches the threshold, a spike occurs, indicating the selection of movie choices. From the active user's movie list, identify the films with the highest ratings. Then, remove all other movies from the list and suggest the highly rated movie to other users. Therefore, the user's movie preference is identified with the help of DCSNN technique.

4. RESULT AND PERFORMANCE EVALUATION

In this section, we present the simulation results of the proposed HFLBC-DCSNN-MRS method and compare its performance against existing methods, namely MT-ML-MRS and GHRS-MRS. Figure 2 illustrates a comprehensive comparison of performance metrics, showcasing the superiority of the HFLBC-DCSNN-MRS approach. Furthermore, Figure 3 presents a comprehensive mean square error (MSE) analysis, showcasing the superior performance of the proposed method across with reduction of the proposed HFLBS-DCSNN-MRS method provides 7%, 4% lesser mean square error for 50 epochs; 6%, 10% lesser mean square error for 100 epoch; 10%, 8% lesser mean square error for 150 epoch; 12%, 6% lesser mean square error for 200 compared to the existing methods for both datasets. The results highlight the efficacy and resilience of the HFLBC-DCSNN-MRS approach in improving classification accuracy and reducing error rates. The proposed method may require enough computational resources due to the integration of deep convolutional spiking neural networks and hesitant fuzzy linguistic clustering. This could be the limition of its applicability in environments with less processing power. As shown in Table 2, the proposed method exhibits significant improvements over the MT-ML-MRS and GHRS-MRS methods.

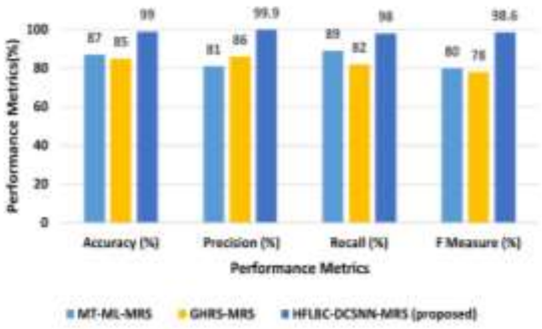

Figure 2. Comparison of various approaches

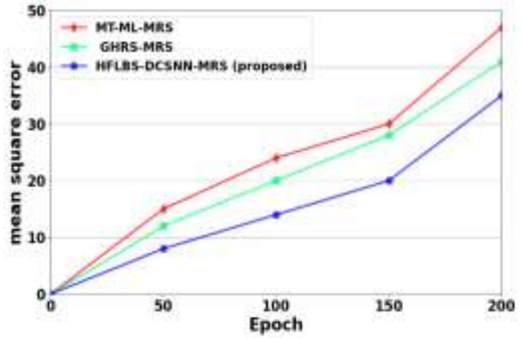

Figure 3. MSE vs epoch

Table 2. Improvement of proposed approach

Metric	Improvement over MT-ML-MRS [19] (%)	Improvement over GHRS-MRS [21] (%)
Accuracy	13.79	16.47
Precision	23.58	16.16
Recall	10.11	19.51
F-measure	23.25	26.41

5. CONCLUSION

The proposed HFLBC along with deep convolution spinking neural network method mentions effective results by combining the strengths of clustering for feature selection and neural networks for classification and recommendation purposes. This combined hybrid approach demonstrates significant improvements over existing approaches such as multi model trust based and graph-based hybrid RS. The performance metrics like accuracy, precision, recall, and F-measure show better results by the proposed method with enhancements of up to 13.79% in accuracy, 23.58% in precision, 10.11% in recall, and 23.25% in F-measure compared to multi trust dependent movie recommender system and surpassed graph-based hybrid recommender system with improvements of 16.47% in accuracy, 16.16% in precision, 19.51% in recall, and 26.41% in F-measure.

In addition, this method shows an effective reduction of mean square error in various epochs in comparison with multi model trust dependent and graph-based hybrid recommender systems. The results show that this system is a better choice than the previous hybrid recommender systems in the case of movie recommendation. The proposed system is not restricted to movies recommendations, as it is a generalized approach. It can be utilized for other application areas like book recommendations andproduct recommendations where user likes and dislikes are associated. Future work can explore the scalability and applicability of our approach on different datasets and domains, paving the way for the development of more complex and effective RS.

REFERENCES

[1] J. W. Bi, Y. Liu, and Z. P. Fan, “A deep neural networks based recommendation algorithm using user and item basic data,” *International Journal of Machine Learning and Cybernetics*, vol. 11, no. 4, pp. 763–777, 2020, doi: 10.1007/s13042-019-00981-y.
[2] S. Salmani and S. Kulkarni, “Hybrid movie recommendation system using machine learning,” in *Proceedings - International Conference on Communication, Information and Computing Technology, ICCICT 2021*, 2021, pp. 1–10, doi: 10.1109/ICCICT50803.2021.9510058.
[3] Q. Shambour, “A deep learning based algorithm for multi-criteria recommender systems,” *Knowledge-Based Systems*, vol. 211, p. 106545, 2021, doi: 10.1016/j.knosys.2020.106545.
[4] M. F. Aljunid and M. Dh, “An efficient deep learning approach for collaborative filtering recommender system,” *Procedia Computer Science*, vol. 171, pp. 829–836, 2020, doi: 10.1016/j.procs.2020.04.090.
[5] D. Roy and M. Dutta, “A systematic review and research perspective on recommender systems,” *Journal of Big Data*, vol. 9, no. 1, p. 59, 2022, doi: 10.1186/s40537-022-00592-5.
[6] Q. Zhang, J. Lu, and Y. Jin, “Artificial intelligence in recommender systems,” *Complex and Intelligent Systems*, vol. 7, no. 1, pp. 439–457, 2021, doi: 10.1007/s40747-020-00212-w.
[7] S. Natarajan, S. Vairavasundaram, S. Natarajan, and A. H. Gandomi, “Resolving data sparsity and cold start problem in collaborative filtering recommender system using linked open data,” *Expert Systems with Applications*, vol. 149, p. 113248, 2020, doi: 10.1016/j.eswa.2020.113248.
[8] E. Zangerle and C. Bauer, “Evaluating recommender systems: survey and framework,” *ACM Computing Surveys*, vol. 55, no. 8, pp. 1–38, 2022, doi: 10.1145/3556536.
[9] F. L. da Silva, B. K. Slodkowski, K. K. A. da Silva, and S. C. Cazella, “A systematic literature review on educational recommender systems for teaching and learning: research trends, limitations and opportunities,” *Education and Information Technologies*, vol. 28, no. 3, pp. 3289–3328, 2023, doi: 10.1007/s10639-022-11341-9.

[10] F. Vahedian, “Weighted hybrid recommendation for heterogeneous networks,” in *RecSys 2014 - Proceedings of the 8th ACM Conference on Recommender Systems*, 2014, pp. 429–432, doi: 10.1145/2645710.2653366.
[11] H. R. Zhang, F. Min, X. He, and Y. Y. Xu, “A hybrid recommender system based on user-recommender interaction,” *Mathematical Problems in Engineering*, vol. 2015, pp. 1–11, 2015, doi: 10.1155/2015/145636.
[12] F. Strub, R. Gaudel, and J. Mary, “Hybrid recommender system based on autoencoders,” in *ACM International Conference Proceeding Series*, 2016, vol. 15-September-2016, pp. 11–16, doi: 10.1145/2988450.2988456.
[13] X. Xu and Y. Zhang, “Collaborative filtering recommendation algorithm based on hybrid similarity,” in *2017 International Conference on Computer Systems, Electronics and Control, ICCSEC 2017*, 2018, pp. 1372–1375, doi: 10.1109/ICCSEC.2017.8447000.
[14] Y. Wang, M. Wang, and W. Xu, “A sentiment-enhanced hybrid recommender system for movie recommendation: a big data analytics framework,” *Wireless Communications and Mobile Computing*, vol. 2018, no. 1, Jan. 2018, doi: 10.1155/2018/8263704.
[15] S. Nagaraj and B. P. Palayyan, “A recommender system-using novel deep network collaborative filtering,” *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 13, no. 1, p. 786, 2024, doi: 10.11591/ijai.v13.i1.pp786-797.
[16] E. Çano and M. Morisio, “Hybrid recommender systems: a systematic literature review,” *Intelligent Data Analysis*, vol. 21, no. 6, pp. 1487–1524, 2017, doi: 10.3233/IDA-163209.
[17] K. R, P. Kumar and B. Bhasker, “DNNRec: a novel deep learning based hybrid recommender system,” *Expert Systems with Applications*, vol. 144, p. 113054, 2020, doi: 10.1016/j.eswa.2019.113054.
[18] H. Tahmasebi, R. Ravanmehr, and R. Mohamadrezaei, “Social movie recommender system based on deep autoencoder network using Twitter data,” *Neural Computing and Applications*, vol. 33, no. 5, pp. 1607–1623, 2021, doi: 10.1007/s00521-020-05085-1.
[19] 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 (Singapore)*, vol. 13, no. 2, pp. 475–482, 2021, doi: 10.1007/s41870-020-00553-2.
[20] V. Shrivastava and S. Kumar, “Comparative analysis of functionality and aspects for hybrid recommender systems,” *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 11, no. 8s, pp. 549–558, 2023, doi: 10.17762/ijritcc.v11i8s.7236.
[21] Z. Z. Darban and M. H. Valipour, “GHRS: graph-based hybrid recommendation system with application to movie recommendation,” *Expert Systems with Applications*, vol. 200, p. 116850, Aug. 2022, doi: 10.1016/j.eswa.2022.116850.
[22] V. Shrivastava and S. Kumar, “Movie recommendation based on fully connected neural network with matrix factorization,” in *Lecture Notes in Electrical Engineering*, vol. 925, 2022, pp. 545–556.
[23] GroupLens, “MovieLens 100K dataset,” *GroupLens*, 2016. https://grouplens.org/datasets/movielens/100k/.
[24] GroupLens and MovieLens, “MovieLens 1M dataset,” *2003*, 2017. https://grouplens.org/datasets/movielens/1m/.
[25] Z. Çataltepe, M. Uluyağmur, and E. Tayfur, “Feature selection for movie recommendation,” *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 24, no. 3, pp. 833–848, 2016, doi: 10.3906/elk-1303-189.
[26] P. Matthews and K. Glitre, “Genre analysis of movies using a topic model of plot summaries,” *Journal of the Association for Information Science and Technology*, vol. 72, no. 12, pp. 1511–1527, 2021, doi: 10.1002/asi.24525.
[27] J. Li, W. Xu, W. Wan, and J. Sun, “Movie recommendation based on bridging movie feature and user interest,” *Journal of Computational Science*, vol. 26, pp. 128–134, 2018, doi: 10.1016/j.jocs.2018.03.009.
[28] Y. Zheng, Z. Xu, Y. He, and Y. Tian, “A hesitant fuzzy linguistic bi-objective clustering method for large-scale group decision-making,” *Expert Systems with Applications*, vol. 168, p. 114355, 2021, doi: 10.1016/j.eswa.2020.114355.
[29] S. Xiang, S. Jiang, X. Liu, T. Zhang, and L. Yu, “Spiking VGG7: deep convolutional spiking neural network with direct training for object recognition,” *Electronics (Switzerland)*, vol. 11, no. 13, p. 2097, 2022, doi: 10.3390/electronics11132097.

BIOGRAPHIES OF AUTHORS

Vineet Shrivastava is a research scholar in the Department of Computer Science and Engineering at Manav Rachna International Institute of Research and Studies, Faridabad, India. He holds a B.Tech. degree in Computer Science and Engineering and an M.Tech. degree in Information Technology. His research focuses on machine learning, particularly recommender systems. He can be contacted at email: vineet.shrivastava62@gmail.com.

Suresh Kumar is a professor in the Department of Computer Science and Engineering at Manav Rachna International Institute of Research and Studies, Faridabad, with over two decades of experience. He is a life member of the Indian Society for Technical Education (ISTE) and the Computer Society of India (CSI) and holds memberships with IEEE and IACSIT, Singapore. His areas of expertise include artificial intelligence, machine learning, and Ad-hoc networks. He has authored over seventy research papers in international journals and conferences. He oversees HelixSmartLabs, a startup incubated at Manav Rachna and registered as a Pvt. Ltd company. He holds two patents and contributed to a technology transfer in the “arules” package of R data analytics software, available from version 1.5.5 onwards at CRAN. He can be contacted at email: enthuvs@gmail.com.