Sentiment Analysis-based Group Recommender System Using Optimization Enabled Random Multi-model Deep Learning For Course Recommendation

[1]Deepjyoti Roy [2*]Shahnawaz Ansari [3]Nazrul Islam

[1]Assistant Professor, Faculty of Engineering, Assam down town University, Guwahati, Assam;
[2]Assistant Professor, Department of Computer Science & Engineering, The Neotia University, West Bengal;
[3]Assistant Professor, Department of Computer Science & Engineering, The Neotia University, West Bengal;

Abstract: Recommender systems are the predominant solution for managing information overload and offering customized suggestions to users for items they may be interested in, such as products, services, or content. This paper presents a sentiment analysis-based recommender system that utilizes an optimization-driven multi-model deep learning approach for course recommendations. The algorithm begins by clustering courses into service groups based on their features. This is achieved by using Fuzzy Local Information C-Means Clustering (FLICM). The user query input is then matched with the service groups using angular distance to determine the most optimal group. Subsequent sentiment classification of the chosen group is conducted using a Random Multi-Model Deep Learning (RMDL) technique. The RMDL model is trained using the proposed hybrid optimization algorithm called Shuffled Shepherd Horse Herd Optimization (SSHHO), which incorporates Shuffled Shepherd optimization algorithm (SSOA) and Horse Herd Optimization (HHO) algorithms. Comparative evaluations with existing methods show that our system achieves enhanced performance in terms of Mean Squared Error (MSE), overall accuracy and Root Mean Squared Error (RMSE), thereby effectively managing information overload and providing tailored course recommendations.

Keywords: Random multi-model deep learning; horse herd optimization; shuffled shepherd optimization; fuzzy local Information C-means clustering.

1. Introduction

A course recommender system suggests educational courses to users based on their interests, preferences, and past behaviour. These systems are commonly used in online learning platforms, educational institutions, and corporate training programs to help users discover relevant courses that align with their goals and learning objectives [1]. After COVID-19, there has been a substantial rise in online learning resources accessible via massive open online courses and learning management systems. In this scenario, the task of personalized course recommendation has become an increasingly significant challenge [2].

Research in the domain of course recommender systems is relatively nascent, and challenging [3]. These limitations include inadequate personalization, limited ability to adapt to user's evolving preferences, and challenges in accurately capturing user feedback. However, sentiment analysis-driven recommender systems, empowered by deep learning techniques, offer a compelling solution [4]. By leveraging the potential of deep learning algorithms to analyse user sentiments expressed in reviews, these systems can extract nuanced insights into user preferences and satisfaction levels. This enables them to offer highly tailored recommendations that resonate with users' individual learning interests and goals [5].

Sentiment analysis-driven recommender systems, empowered by deep learning techniques, offer a compelling solution by effectively harnessing the extensive volumes of unstructured data in user reviews and feedback to improve recommendation accuracy and personalization. Recent studies concentrate on deep learning models such as Convolutional Neural Networks (CNN) [6], Long Short-Term Memory (LSTM) [7], Bi-LSTM [8], and Bidirectional Encoder Representations from Transformers (BERT) [9]. These techniques excel at capturing nuanced sentiments and emotions expressed in textual data, thereby providing more reliable insights into user preferences. Additionally, researchers have introduced various approaches to enhance the performance of course recommender systems using free-text reviews. These approaches typically involve topic modeling [10], which automatically extracts features that are then combined with the latent factor model to predict ratings. Topic modeling techniques identify hidden topics within reviews, providing a more nuanced understanding of user preferences and course attributes. By integrating these extracted features with latent factor models, which capture underlying user-item interactions, the systems can make more accurate and personalized recommendations. Also, many recommender models incorporate clustering schemes to further improve performance. Clustering techniques group users or items based on historical ratings or other auxiliary data, such as demographic information or behavioral patterns. Methods like k-means clustering and hierarchical clustering help in identifying similar users or items, which can be particularly useful for collaborative filtering approaches [11]. By recognizing similar neighbors, the system can leverage their preferences to generate recommendations for a target user, even in the absence of extensive rating data for that user.

Despite these advancements, numerous challenges persist in developing a course recommender system. A major issue is the cold start problem, where the recommender system has difficulty providing efficient recommendations for new users due to insufficient data. Furthermore, the complexity and computational expense of deep learning models can be prohibitive, especially for real-time applications. Another issue is the potential bias in the data, which can lead to skewed recommendations that do not accurately reflect user preferences. Furthermore, while sentiment analysis can provide valuable insights, it is often hampered by the ambiguity and variability of natural language, making it difficult to achieve consistently high accuracy. Topic modeling can sometimes struggle with the high dimensionality and sparsity of textual data, leading to less accurate feature extraction. Moreover, the integration of topic features with latent factor models can increase computational complexity, making it difficult to scale these systems for large datasets. Clustering techniques, while effective, may also face issues with the heterogeneity of user data, where diverse user behaviors and preferences complicate the identification of similar groups [11].

This paper introduces a sentiment analysis-based recommender system that leverages a multi-model deep learning approach optimized for course recommendations. The algorithm initiates the process by clustering courses into service groups based on their features using FLICM [12]. User queries are then matched with these service groups through angular distance calculations to identify the most relevant group. Following this, sentiment analysis of the selected group is performed using RMDL technique [13]. The RMDL model's training is enhanced by the proposed hybrid optimization algorithm, Shuffled Shepherd Horse Herd Optimization (SSHHO), which combines the strengths of Shuffled Shepherd Optimization Algorithm (SSOA) [14] and Horse Herd Optimization (HHO) [15].

The system's effectiveness is demonstrated through comparative evaluations against existing methods, showcasing superior performance in terms of MSE, RMSE and overall accuracy. This innovative approach not only manages information overload more efficiently but also delivers more precise and personalized course recommendations.

By integrating advanced clustering, sentiment analysis, and hybrid optimization techniques, the system provides a robust framework to enhance the accuracy of recommendations. The clustering of courses into service groups ensures that user queries are matched with the most pertinent categories, while the sentiment analysis fine-tunes the recommendations based on user feedback. The hybrid optimization algorithm, SSHHO, enhances the learning process of the RMDL model, leading to more accurate predictions and better handling of diverse user preferences. Overall, this comprehensive approach addresses key challenges in recommendation systems, such as information overload and user satisfaction, thereby offering a significant improvement over traditional methods.

The rest of the paper is organized as follows: In Section 2, a review of some recent works on various recommender systems based on sentiment analysis is provided. Section 3 outlines a description of the proposed methodology. Section 4 describes the advantages of the proposed system over some recent existing systems, highlighting its strengths and innovations. Section 5 presents a comparative analysis of the proposed system against traditional methods, demonstrating its superior performance. Finally, Section 6 and 7 offer the conclusion and future scope of research, respectively, summarizing the findings and suggesting directions for further research.

2. Literature Survey

2.1. Recent related works

In 2023, in [4], a Light Deep Learning (LightDL)-based recommender system that applies sentiment analysis to enhance recommendation accuracy was proposed. The system processes Twitter-based reviews, extracting semantic, syntactic, symbolic, and tweet-based features. The model classifies sentiments into positive sentiments, negative sentiments and neutral sentiment categories, providing tailored recommendations based on user feedback. The LightDL model demonstrates superior performance across various metrics, achieving 95% accuracy on the Twitter dataset. [16] introduces SARWAS, a deep ensemble learning method that integrates sentiment analysis with deep neural networks for recommendation systems. The system processes both ratings and reviews, using sentiment analysis to convert textual feedback into numerical values. These inputs are fed into an ensemble of deep learning architectures to improve learning capabilities. The system outperforms traditional methods in precision, loss, and execution time, highlighting the effectiveness of combining multiple deep learning models. In [17], the authors apply deep learning techniques to analyze sentiment in TripAdvisor reviews, providing insights for hotel recommendations. The study utilizes BERT for review classification based on sentiment. The model's effectiveness is demonstrated through comparative analysis with other deep learning models, making it a robust method for understanding customer feedback and improving service recommendations. [18] presents a hybrid course recommender system combining content-based and collaborative filtering approaches. The system uses RNNs and LSTMs to overcome the shortcomings of traditional methods. By treating recommendations as a sequential problem, the system improves personalization and accuracy. The proposed method outperforms other systems in various performance metrics, demonstrating the potential of hybrid deep learning models for course recommendations.

In 2022, [19] introduced a novel sentiment deep learning technique for multi-criteria recommendation systems. The proposed method combines auto encoders with sentiment analysis models to process multi-criteria feedback. The system demonstrates improved performance on the TripAdvisor dataset, outperforming existing methods by leveraging sentiment information to enhance recommendation accuracy. [20] compares clustering and topic modeling techniques for recommendation systems. The authors implement clustering methods to group similar items and use topic modeling to extract features from textual data.

2.2. Challenges

In recent years, recommender systems have been the focus of extensive research, with a growing body of literature exploring the use of sentiment analysis across a variety of domains. However, there are several significant challenges that need to be addressed to build an efficient recommender system. Some of the challenges are as follows:

- Data Quality and Quantity:

  Building a reliable sentiment analysis-based recommender system requires large amounts of high-quality data. This means that there must be an extensive collection of user reviews, ratings, and feedback that can be used to train the machine learning model [21]. Furthermore, the data quality can impact the accuracy of the sentiment analysis, which can negatively impact the recommendations.
- Bias:

  Sentiment analysis-based recommender systems can be biased, particularly if the data utilized for train the model is biased. This can lead to inaccurate recommendations and can negatively impact user experience [22].
- Explainability:

  Recommender systems that use sentiment analysis can be complex and difficult to build. More focus is necessary to develop transparent and explainable models that can help users understand how recommendations are generated [23].
- Personalization:

  Sentiment analysis-based recommender systems should be personalized to each user's unique needs and interests. This requires gathering detailed information about the user, such as their learning goals, preferred learning style, and previous educational experiences. Incorporating this information into a sentiment analysis-based recommender system can be challenging, as it requires sophisticated algorithms that can process large amounts of data [24].

- User Privacy:
  Recommender systems that use sentiment analysis must be designed to protect user privacy. This requires careful attention to data collection and analysis, as well as the development of privacy-preserving algorithms that can generate accurate recommendations without compromising user privacy [25].

3. Proposed methodology

This section outlines the proposed methodology for the sentiment analysis-based course recommender system. The process involves three main phases viz., clustering, group matching, and sentiment classification.

In the initial phase, courses are clustered into groups, referred to as service groups, based on their features using FLICM clustering. FLICM enhances traditional fuzzy c-means clustering by incorporating local spatial information to enhance clustering accuracy and robustness against noise. Each course is represented as a feature vector encompassing attributes such as course content, duration, difficulty level, and user ratings. FLICM provides membership levels to each course for each cluster, reflecting the degree of association with each cluster. The algorithm iteratively updates these membership levels and cluster centers by minimizing an objective function that considers both the distance and local spatial information. This clustering process results in the formation of distinct service groups, each containing courses with similar characteristics, thus facilitating personalized recommendations.

Once the courses are clustered, the next step involves matching the user query input with the service groups to determine the most optimal group. This matching is performed using angular distance, a metric that finds the cosine of the angle between the user query vector and the centroid vectors of the service groups. Angular distance is particularly effective in high-dimensional spaces and is less sensitive to the magnitude of the vectors, focusing instead on their orientation. The service group with the smallest angular distance to the user query is selected as the most relevant group, ensuring that the courses recommended to the user are closely aligned with their preferences and requirements.

After identifying the optimal service group, sentiment analysis of the user reviews within the courses of this group is conducted using a RMDL technique. RMDL employs an ensemble of multiple deep learning models, each randomly initialized and trained to capture diverse aspects of the data. This approach enhances the robustness and accuracy of the sentiment classification process. The RMDL model is trained using a proposed hybrid optimization algorithm known as SSHHO, which combines the SSOA and HHO algorithms to optimize the training process effectively. SSOA contributes by enhancing the exploration capabilities of the optimization, while HHO improves exploitation, ensuring a balanced search and convergence towards optimal solutions. The sentiment classification results provide valuable insights into user satisfaction and preferences, further refining the recommendation process. The schematic diagram of the proposed course recommender system is illustrated in Figure 1.

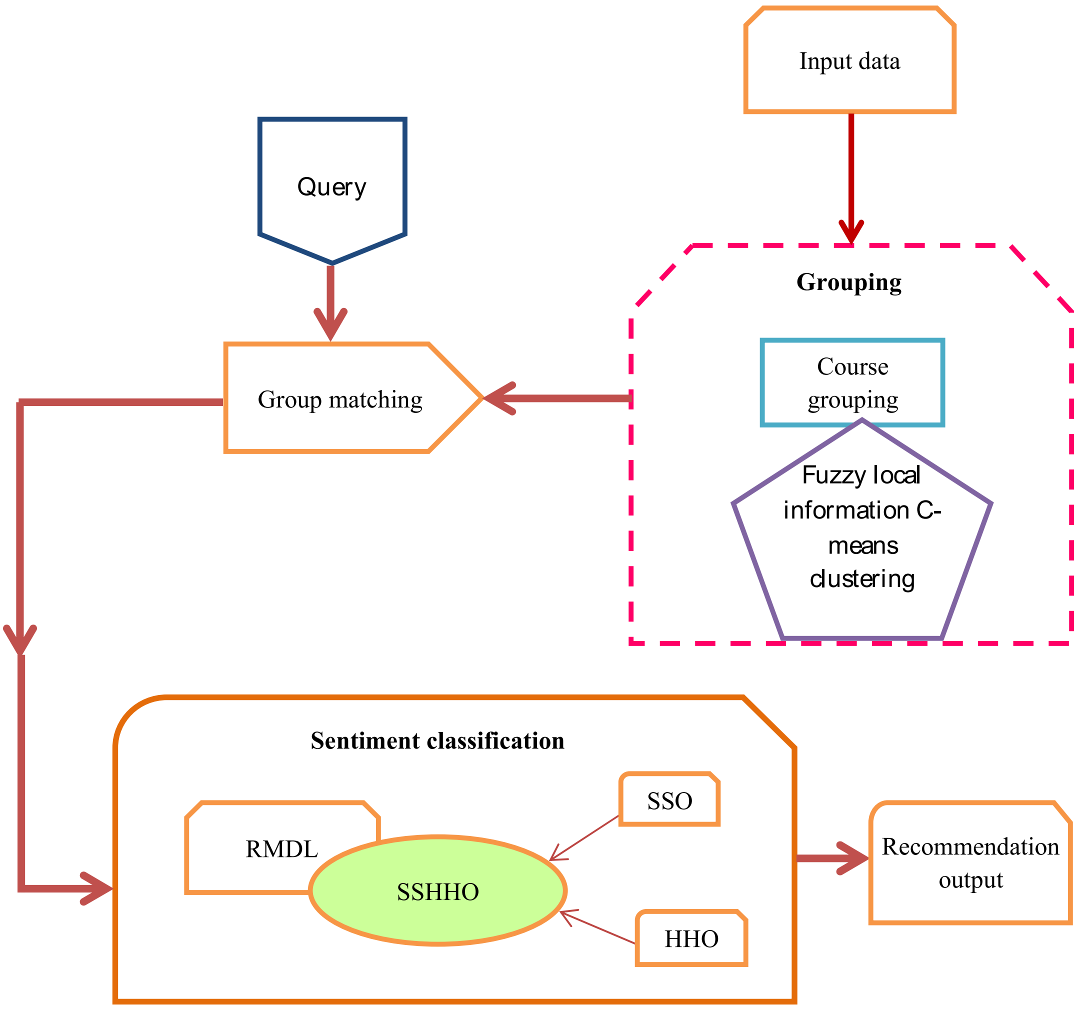


Figure 1. Schematic diagram of the proposed course recommender system.

To evaluate the effectiveness of the proposed course recommender system, comparative analyses with existing methods are performed. Metrics such as MSE, accuracy and RMSE are used to assess the system's performance. The evaluations demonstrate that our system achieves enhanced performance across these metrics, indicating its superiority in providing accurate and reliable course recommendations. The integration of FLICM for clustering, angular distance for query matching, and RMDL with SSHHO for sentiment classification collectively contribute to the system's improved efficacy and user satisfaction.

3.1. Dataset Description

The Coursera course dataset [26] from Kaggle was utilized as the dataset for the proposed model. The dataset consists of 888 different courses.

3.2. Course grouping using FLICM

FLICM clustering is an advanced clustering technique that enhances the traditional Fuzzy C-Means (FCM) algorithm by integrating local spatial information to improve clustering accuracy and robustness against noise. This method is particularly effective in scenarios where the data contains spatial correlations, making it well-suited for clustering courses based on their features.

3.2.1 Course Representation

Each course in our dataset is represented as a feature vector. The features include:

- Course Type: Encoded as a vector of keywords or topics covered.
- Duration: The length of the course in hours.
- Course Difficulty Level: Categorical variable encoded numerically (e.g., Beginner = 1, Intermediate = 2, Advanced = 3).
- Course Ratings: Average user rating on a scale from 1 to 5.
- Students enrolled or Subscription Data: Number of users subscribed to the course.

These feature vectors form the input for the FLICM clustering process.

3.2.2. FCILM

The FLICM algorithm begins by initializing the cluster centres and membership values, similar to the traditional FCM. The key difference in FLICM is the inclusion of a local spatial term in the objective function, which accounts for the spatial relationship between data points. The objective function for FLICM is defined as:

$$J_m = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ij}^m \|x_i - v_j\|^2 + \lambda \sum_{i=1}^{N} \sum_{j=1}^{C} \sum_{k \in N_i} \frac{u_{ij}^m}{1+\|x_i - x_k\|^2}$$

J_m is the objective function,

N depicts the number of courses,

C denotes the number of clusters,

u_{ij} is the membership degree of course x_i in cluster j,

m signifies the fuzziness parameter,

x_i and x_k are data points,

v_j is the cluster centre of cluster j,

λ is the weighting parameter for the spatial term,

N_i denotes the local neighbourhood of data point x_i.

- Initialization: The cluster centres v_j for j = 1,2,3, ..., C are first initialized randomly. Then the membership matrix U is initialized, such that $u_{ij} \in [0,1]$ and $\sum_{j=1}^{C} uij = 1$ for all i.
- Membership update: Update the membership values using the formula:

$$u_{ij} = \left(\sum_{k=1}^{C} \left(\frac{\|x_i - v_j\|^2 + \lambda \sum_{k \in N_i} \frac{1}{1+\|x_i - x_k\|^2}}{\|x_i - v_k\|^2 + \lambda \sum_{k \in N_i} \frac{1}{1+\|x_i - x_k\|^2}} \right)^{\frac{1}{m-1}} \right)^{-1}$$

- Cluster center update: The cluster centers are then updated using the formula:

$$v_j = \frac{\sum_{i=1}^{N} u_{ij}^m x_i}{\sum_{i=1}^{N} u_{ij}^m}$$

The FLICM algorithm continuously updates the membership values and cluster centers until it reaches convergence. Convergence is typically accomplished when the change in the objective function J_m or the change in membership values between iterations falls below a predefined threshold.

3.2.3. Implementation of FCILM

The implementation of FLICM for clustering courses includes the following steps:

1. Data preprocessing: First, we normalize the feature vectors to make certain that all features equally contribute to the distance calculations. Then we define the neighborhood Ni for each data point, typically based on the k-nearest neighbors.
2. Parameter selection: We set the fuzziness parameter m (here we take m=2) and choose the weighting parameter λ to balance the spatial term's influence.
3. Algorithm Execution: The membership degree and cluster centers are initialized. Then, the membership values and cluster centers are repeatedly updated until convergence is achieved. We assign each course to the cluster with the highest membership value. At the end of this step we are left with distinct service groups, each containing courses with similar characteristics.

3.3. Matching User Query with Service Groups Using Angular Distance

Angular distance is a metric used to measure the orientation of two vectors in a high-dimensional space. It is particularly useful for comparing the similarity between data points regardless of their magnitude, focusing instead on the direction. This makes it well-suited for matching user queries with clustered service groups in a course recommender system, where the query and clusters are represented as high-dimensional feature vectors.

3.3.1. User Query Representation

When a user searches for a course, the query is represented as a feature vector like the representation used for clustering courses. The feature vector for the user query includes:

- Keywords or Topics: Represented as a vector of keywords/topics covered in the course.
- Duration: Length of the course in hours.
- Difficulty Level: Encoded numerically (e.g., Beginner = 1, Intermediate = 2, Advanced = 3).
- User Ratings: The average rating of the course.
- Subscription Data: Number of users subscribed to the course.

3.3.2. Matching Process Using Angular Distance

The angular distance between vectors A and vector B is given by the cosine similarity measure, which is calculated as:

$$\text{cosine\_similarity}\ (A\ ,\ B) = \frac{A.B}{||A||\ ||B||}$$

where, A.B represents the dot product of vector A and B while ||A|| and ||B|| signifies the magnitudes of vector A and B. The angular distance can then be derived from the cosine similarity as:

$$\text{angular\_distance}\ (A, B) = 1 - \text{cosine\_similarity}\ (A, B)$$

3.3.3. Implementation

1. Vector Representation: The user query and the centroid vectors of the service groups are converted into their respective feature vectors.
2. Calculate Cosine Similarity: Calculate the cosine similarity between the user query vector and each of the centroid vectors of the service groups. Let us assume that a user searches for a course with a feature vector Q, and we have service group centroids $C_1, C_2, ..., C_n$. Then the cosine similarity for each centroid is computed as follows:

$$\text{cosine\_similarity}\ (Q, C_i) = \frac{Q, Ci}{||Q||\ ||Ci||}\ , \forall i = 1, 2, ..., n.$$

3. Determine Angular Distance: Convert the cosine similarity to angular distance using the formula:

$$\text{angular\_distance}\ (Q, C_i) = 1 - \text{cosine\_similarity}\ (Q\ , C_i)\ , \forall\ i = 1, 2, ..., n.$$

4. Identify the Optimal Group: Finally, the service group with the smallest angular distance to the user query is selected.

$$\text{Optimal Group} = \min\ (\text{angular\_distance}\ (Q, C_i))$$

This group now called as service group, represents the cluster of courses that are most relevant to the user's search, is now forwarded to the sentiment classification phase.

3.4. Sentiment Classification Using Random Multi-Model Deep Learning

Sentiment classification involves analysing the user reviews of the courses within the derived service group to find out the expressed sentiment, typically classified as positive sentiment, negative sentiment, or neutral sentiment. In the context of a course recommender system, sentiment classification offers valuable insights into user satisfaction and course quality, enhancing the recommendation process. We employ a RMDL technique for robust sentiment analysis, leveraging multiple deep learning models to improve classification accuracy and reliability.

3.4.1. Random Multi-Model Deep Learning (RMDL)

RMDL involves training an ensemble of different deep learning models, each initialized randomly and trained on the same dataset. The ensemble approach captures diverse patterns and features from the data, improving the overall performance and robustness of the sentiment classification. The RMDL framework typically includes models such as Recurrent Neural Networks (RNNs), CNNs, and Deep Neural Networks (DNNs). The structure of RDML is given in Figure 2.

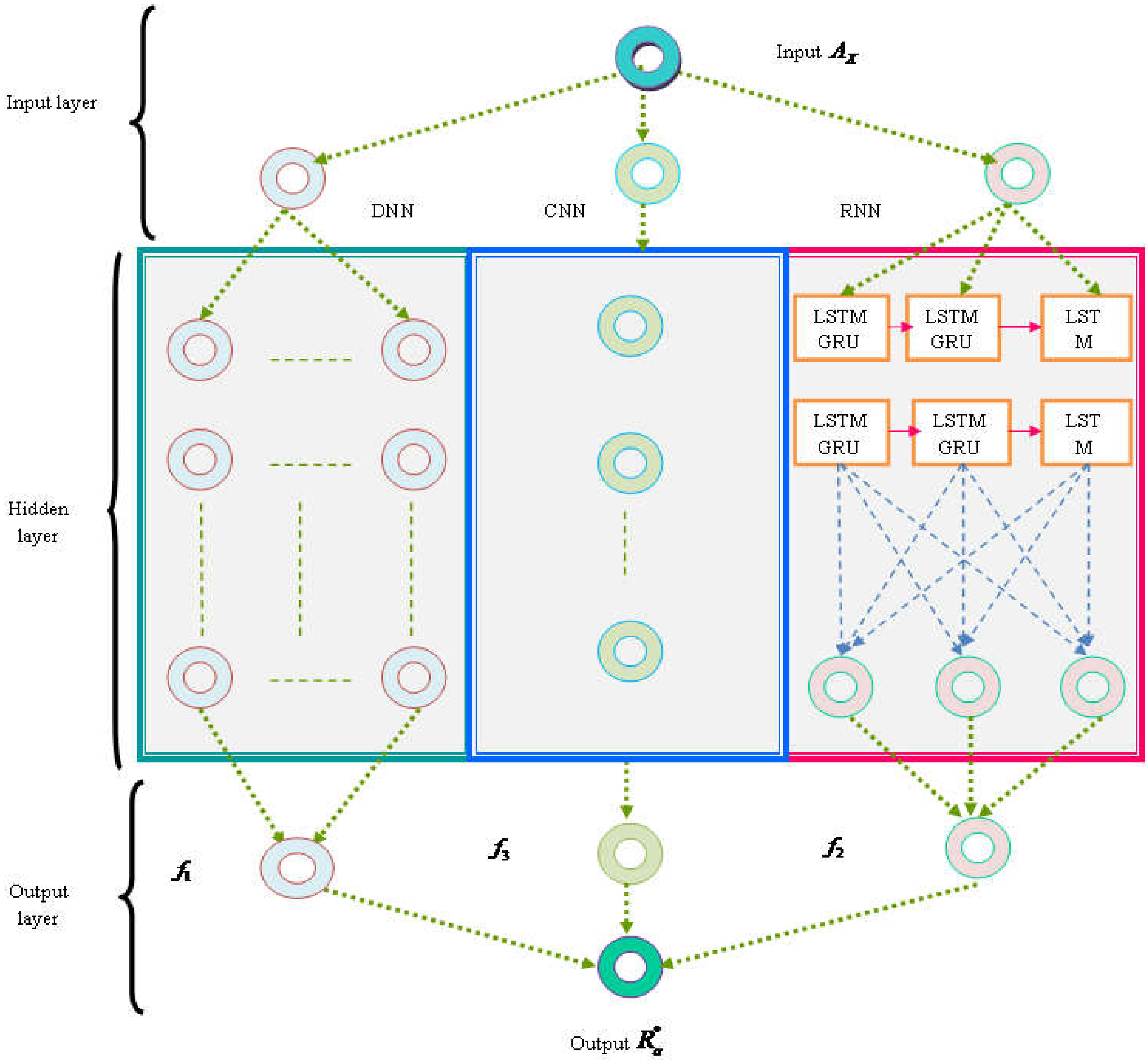


Figure 2. Structure of RMDL model

3.4.2. Components of RMDL

1. RNNs: RNNs are highly effective at processing sequential data, making them appropriate for analyzing temporal dependencies in text data. The hidden state of the RNN is updated at each time step *t* based on the input x_t and the previous hidden state h_{t-1}:

$$h_t = \sigma\,(W_h\, h_{t-1} + W_x\, x_t + b)$$

where, W_h and W_x represents weight matrices, b denotes the bias, and σ signifies an activation function.

2. CNNs: CNNs can capture local patterns in text data through convolutional operations. A convolutional layer applies a set of filters to the input text, generating feature maps:

$$f_k = \sigma\,(W_k * x + b_k),$$

where, W_k is the filter, ∗ represents the convolution operation, x is the input, and b_k is the bias term.

3. DNNs: DNNs consist of several layers of neurons, allowing them to model complex non-linear relationships in the data. The output of each layer is computed as:

$$h^{(l)} = \sigma(W^{(l)}\, h^{(l-1)} + b^{(l)})$$

where, $h^{(l-1)}$ denotes the input from the previous layer, $W^{(l)}$ represents the weight matrix, $b^{(l)}$ signifies the bias, and σ is an activation function.

3.4.3. Training Process of RMDL using Shuffled Shepherd Horse Herd Optimization (SSHHO)

The training process of RMDL involves the following key steps:

1. Data Preparation:
   - Preprocessing: User reviews are preprocessed to remove stop words and punctuation. Text is then tokenized and transformed into numerical form using techniques such as word embeddings.
   - Labeling: Reviews are labeled based on their sentiment, typically as positive sentiment, negative sentiment, or neutral sentiment.
2. Model Initialization:
   Multiple deep learning models (e.g., CNNs, RNNs, LSTMs) are randomly initialized. Each model captures different aspects of the review data, contributing to the ensemble's overall performance.
3. Hybrid Optimization with SSHHO:
   - SSOA: Enhances the exploration capabilities of the optimization process. It divides the population into subgroups (shepherds) that explore different regions of the search space.
   - Horse Herd Optimization (HHO): Focuses on exploitation, refining the search around promising areas identified by the shepherds. It mimics the social behavior of horse herds to optimize solutions effectively.
   - Hybrid SSHHO: Combines the strengths of SSOA and HHO to balance exploration and exploitation. The objective is to optimize the parameters of the deep learning models to improve their performance on sentiment classification.

3.4.4. Implementation of SSHHO:

The implementation process of SSHHO is described as follows:

1. Implementation of SSOA
   - First, we initialize the population (shepherds) and evaluate their fitness.
   - Next, we divide the population into subgroups.
   - Each subgroup explores different regions of the search space.
   - The position of every shepherd is updated based on the equation:

$$X_i^{t+1} = X_i^t + \alpha \cdot (X_{best}^t - X_i^t) + \beta \cdot (X_{random}^t - X_i^t)$$

where, X_i^t is the position of the i[th] shepherd at iteration t, X_{best}^t is the best solution found, X_{random}^t is the random solution and α and β are control parameters.

2. Implementation of HHO:
   - First, we initialize herd population and evaluate fitness.
   - Horses adjust their positions based on social hierarchy and best-found solutions.
   - The position of each horse is updated using the formula:

$$X_i^{t+1} = X_i^t + \gamma \cdot (X_{best}^t - X_i^t)$$, where, γ is the learning rate.

3. Implementation of SSHHO:
   - First, we integrate SSOA and HHO by alternating between exploration (SSOA) and exploitation (HHO) phases.
   - Balanced optimization is ensured to fine-tune the deep learning models effectively.

3.4.5. Model Training

Each deep learning model in the RMDL ensemble is trained using the SSHHO-optimized parameters. The models are trained on the preprocessed and labeled review data, learning to classify the sentiment of reviews.

3.4.6. Integration

At last, we combine the outputs of all trained models in the RMDL ensemble. The final sentiment classification is determined using majority voting. The courses within the service group with positive sentiments are finally recommended to the user.

3.4.7. Pseudo code of the proposed SSHHO

The SSHHO algorithm combines the SSOA and HHO to balance exploration and exploitation for effective optimization. The pseudo code for the SSHHO algorithm is as follows:

```
Initialize the population of horses and shepherds (candidate solutions)
Access the fitness of each candidate solution
Divide the population into subgroups (shepherds) based on their fitness
WHILE (termination criteria not met) DO
    // SSOA Phase: Exploration
    FOR each subgroup DO
        Identify the best solution (shepherd) in the subgroup
        FOR each candidate solution in the subgroup DO
            Update position using: X_i^(t+1) = X_i^(t) + alpha * (X_best^(t) - X_i^(t)) +
            beta * (X_random^(t) - X_i^(t))
            Evaluate the new position
        ENDFOR
    ENDFOR

    // HHO Phase: Exploitation
    Identify the global best solution (X_best)
    FOR each candidate solution (horse) DO
        Update position using: X_i^(t+1) = X_i^(t) + gamma * (X_best^(t) - X_i^(t))
        Evaluate the new position
```

ENDFOR

// Shuffle Phase: Diversity Introduction

Shuffle candidate solutions to introduce diversity

Re-divide population into new subgroups

Update global best solution if improved

ENDWHILE

Return the global best solution

4. Advantages of the proposed system using RMDL with SSHHO

The proposed system leverages Random Multi-Model Deep Learning (RMDL) trained with the Shuffled Shepherd Horse Herd Optimization (SSHHO) algorithm to address the limitations of existing sentiment classification systems in course recommender systems. Below, we detail the advantages of this novel approach compared to traditional systems and explain how it outperforms and overcomes the disadvantages of existing models.

1. Enhanced robustness and generalization:

Existing Systems: Traditional sentiment classification models often rely on a single deep learning model or a simple ensemble of homogeneous models, which can lead to overfitting and poor generalization, especially when dealing with diverse and noisy data [27-29].

Proposed System: RMDL employs an ensemble of multiple deep learning models (e.g., CNNs, RNNs, LSTMs), each randomly initialized and trained differently. This diversity allows the system to capture various patterns and aspects of the review data, significantly enhancing robustness and generalization. The ensemble approach mitigates the risk of overfitting by balancing the strengths and weaknesses of individual models.

2. Superior optimization with SSHHO:

Existing Systems: Optimization of model parameters in traditional systems is often done using basic algorithms like Stochastic Gradient Descent (SGD) [30], Adam [31], or other standard techniques [32]. These methods may struggle with complex optimization landscapes, leading to sub-optimal solutions and slower convergence.

Proposed System: SSHHO combines the strengths of SSOA and HHO, providing a balanced approach to exploration and exploitation. SSOA enhances exploration by dividing the population into subgroups that explore different regions, while HHO focuses on exploitation by refining solutions based on social hierarchy and best-found solutions. This hybrid approach ensures faster convergence to optimal solutions and more effective training of the deep learning models.

3. Mathematical Advantage:

The SSHHO algorithm is designed to navigate complex optimization landscapes more effectively than traditional optimization techniques. It minimizes the risk of getting stuck in local minima by dynamically balancing exploration and exploitation:

$$X_i^{t+1} = X_i^t + \alpha \cdot (X_{best}^t - X_i^t) + \beta \cdot (X_{random}^t - X_i^t)$$

This formula combines the benefits of random exploration and directed exploitation, leading to more robust parameter optimization.

4. Improved Performance Metrics:

Existing Systems: Traditional systems may not consistently achieve high performance across various metrics such as MSE, RMSE and accuracy due to the limitations in model diversity and optimization techniques [33].

Proposed System: The combination of RMDL and SSHHO significantly enhances performance metrics. By leveraging multiple models and an advanced optimization technique, the proposed system consistently achieves reduced MSE and RMSE values and increased accuracy. This improvement is evident in the system's ability to accurately classify sentiments in diverse and noisy review datasets.

5. Handling Data Complexity and Noise

Existing Systems: Single model approaches or simple ensembles often struggle with complex and noisy datasets, leading to reduced classification accuracy and reliability [34-37].

Proposed System: RMDL's ensemble of diverse models captures various data complexities, while SSHHO's robust optimization ensures that each model is finely tuned to handle noise and outliers effectively. This results in a more accurate and reliable sentiment classification.

6. Versatility and Scalability

Existing Systems: Traditional systems often lack versatility and scalability, struggling to adapt to new data or scale effectively with increasing data volume [38-41].

Proposed System: The modular nature of RMDL allows for easy adaptation to new datasets by training additional models or retraining existing ones. SSHHO's efficient optimization process supports scalability, enabling the system to handle large volumes of data without significant performance degradation.

5. Results and Discussion

The results obtained from the proposed course recommender system is discussed in this section.

5.1. Experimental Setup

The experimental setup involved a model developed and tested using the Python programming language. It was executed on a computer powered by a 2.7 GHz Dual-Core Intel Core i5 processor, running the MACOS operating system.

5.2. Performance Metrics

The system's performance was evaluated using various metrics, including MSE, RMSE, and accuracy. These metrics measured the model's accuracy in predicting course recommendations.

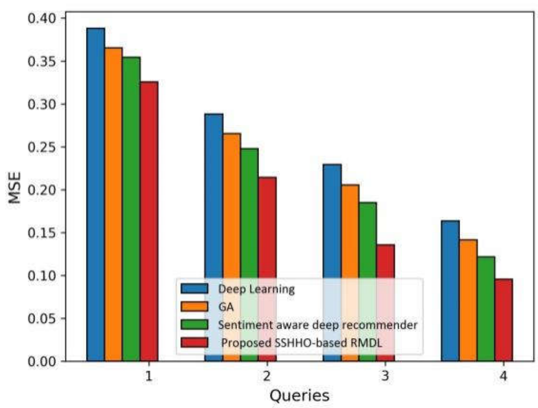
- MSE: MSE evaluates the average squared differences between the predicted and actual values. A lower MSE value signifies better model performance.
- RMSE: RMSE, which is the square root of the MSE, quantifies the magnitude of the error between the predicted and actual values. It also indicates how well the model fits the data. A lower RMSE value signifies better model performance. Mathematically it can be represented as: RMSE = $(E_e)^{1/2}$ where, E_e signifies the MSE values.
- Accuracy: Testing accuracy evaluates the proportion of accurate predictions made by the model on a test data set. It is determined by the ratio of correct predictions to the total number of predictions. A higher testing accuracy value indicates superior model performance.

5.3. Comparative Analysis

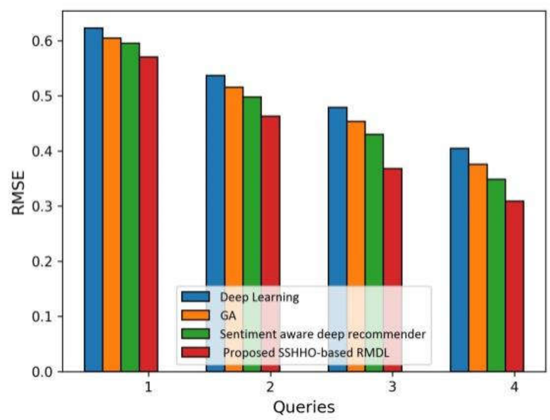
The efficacy of the proposed system was evaluated by comparing it with various conventional techniques, including deep learning (DL) [42], genetic algorithms (GA) [43], and the Sentiment-Aware deep recommender system (SADRS) [44]. Assessment was done with cluster size 3 and 4.

5.3.1. Assessment with cluster size = 3

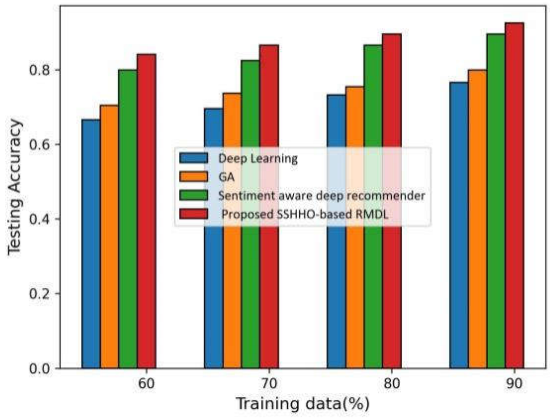
Figures 3(a) and 3(b) show the comparative analysis in terms of MSE and RMSE respectively, across various iterations for four different user queries. Figure 3(c) presents the comparative analysis in terms of accuracy calculated with varying percentages of the training set data.



(a)



(b)

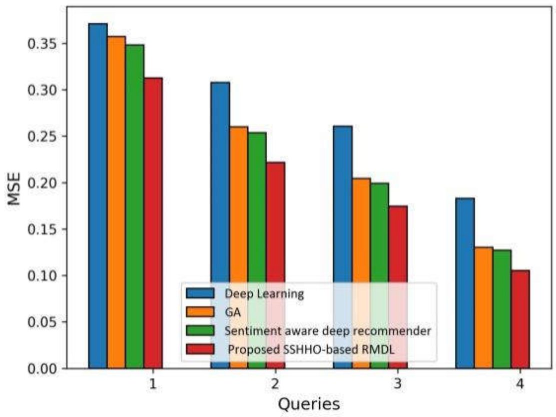


(c)

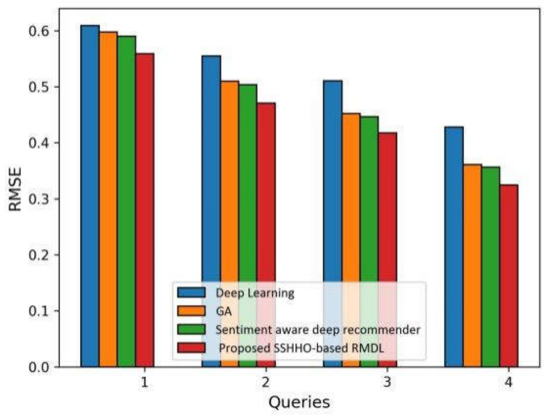
Figure 3. Assessment of devised model based on a) MSE, b) RMSE and c) Accuracy

5.3.1. Assessment with cluster size = 4

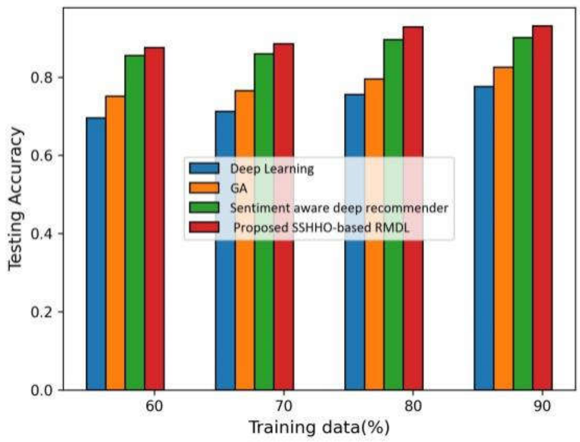
Figures 4(a) and 4(b) show the comparative analysis in terms of MSE and RMSE, respectively, across various iterations for four different user queries. Figure 4(c) presents the comparative analysis in terms of accuracy calculated with varying percentages of the training set data.



(a)



(b)



(c)

Figure 4. Assessment of devised model based on a) MSE, b) RMSE, and c) Testing accuracy

Table 1 describes a comparative analysis of the proposed system with DL [42], GA [43], and SADRS [44] in a Tabular format.

Table 1. Comparative Analysis

Variations	Metrics	DL	GA	SADRS	Proposed System
Cluster size=3	MSE	0.1634	0.1412	0.1214	**0.0954**
	RMSE	0.4042	0.3758	0.3484	**0.3088**
	Accuracy	0.7654	0.7985	0.8954	**0.9254**
Cluster size=4	MSE	0.1831	0.1301	0.1269	**0.1052**
	RMSE	0.4279	0.3608	0.3562	**0.3244**
	Accuracy	0.7754	0.8254	0.9014	**0.9314**

6. Conclusion

In this research, we introduced a novel sentiment analysis-based group recommender system for course recommendations, leveraging advanced clustering and deep learning techniques to enhance recommendation accuracy and user satisfaction. Our approach begins with the use of FLICM to group courses into service groups based on their features, ensuring that courses with similar characteristics are grouped together. This is followed by matching user queries with these service groups using angular distance, a metric that effectively captures the orientation of high-dimensional vectors, thus ensuring precise alignment between user preferences and course attributes. The core of our system is the sentiment classification of user reviews using a Random Multi-Model Deep Learning (RMDL) technique, optimized through the proposed Shuffled Shepherd Horse Herd Optimization (SSHHO) algorithm. This hybrid optimization technique enhances both the exploration and exploitation phases of model training, leading to superior performance metrics such as MSE, RMSE and accuracy. Comparative evaluations demonstrate that our system outperforms traditional methods by addressing their limitations in model diversity and optimization. Consequently, our proposed system not only provides more accurate and reliable course recommendations but also significantly improves user engagement and satisfaction, marking a substantial advancement in the field of recommender systems.

7. Future Scope

The proposed system marks a significant advancement in the domain of course recommendation systems. This innovative approach overcomes several key challenges inherent in traditional recommender systems, such as limited model diversity, sub-optimal optimization techniques, and poor generalization to diverse and noisy datasets. By integrating advanced clustering methods and hybrid optimization techniques, our system achieves superior accuracy, robustness, and user satisfaction. Looking forward, there are numerous avenues for further research and development. Future work could explore the incorporation of additional contextual information, such as user learning styles and real-time engagement metrics, to further personalize recommendations. Moreover, incorporating adaptive learning mechanisms that evolve based on user feedback could enhance the system's adaptability and responsiveness. Investigating the application of this framework to other domains beyond education, such as personalized healthcare or e-commerce, presents another exciting direction. Leveraging explainable AI techniques to make the recommendation process more transparent could build user trust and provide deeper insights into the decision-making process. This research establishes a robust foundation for the ongoing development of intelligent, user-focused recommender systems, promising to catalyze additional innovations and applications within the field.

Acknowledgment

We would like to express our gratitude to the following organizations and individuals for their invaluable support in completing this research project.

First of all, we would like to express our sincere gratitude to Dr. Kapil K. Nagwanshi for his support and guidance throughout this research. The insights and feedback have been helpful in shaping the scope and directions of this study. We are also grateful to Prof. Partha Mukherji, Dr. Manas Paul for their assistance with data collection, analysis, and interpretation, which made the study possible. We would also like to express our gratitude to the members of Research and Development Committee for their constructive criticism and valuable feedback, which contributed to enhancing the quality of this research.

Finally, we thank The Neotia University for providing the necessary facilities and resources to conduct our research. We extend our gratitude to all for your invaluable contributions that significantly enhanced the quality of the paper.

References

1. Algarni, S.; Sheldon, F. Systematic Review of Recommendation Systems for Course Selection. *Machine Learning and Knowledge Extraction*, **2023**, 5(2), 560-596.
2. Vedavathi, N.; KM, A. K. E-learning course recommendation based on sentiment analysis using hybrid Elman similarity. *Knowledge-Based Systems*, **2023**, 259, 110086.
3. George, G.; Lal, A. M. Review of ontology-based recommender systems in e-learning. *Computers & Education*, **2019**, 142, 103642.
4. Chiranjeevi, P.; Rajaram, A. A lightweight deep learning model-based recommender system by sentiment analysis. *Journal of Intelligent & Fuzzy Systems*, **2023**, 1-14.
5. Smitha, E. S., Sendhilkumar, S.; Mahalakshmi, G. S. Intelligence system for sentiment classification with deep topic embedding using N-gram based topic modeling. *Journal of Intelligent & Fuzzy Systems*, **2023**, 1-27.

6. Sadouni, O.; Zitouni, A. New Recommendation System Based on Students' Engagement Prediction Using CNN to Optimize E-Learning. *International Journal of Organizational and Collective Intelligence*, **2022**, 12(4), 1-27.
7. Deepak, G.; Trivedi, I. A Hybridized Deep Learning Strategy for Course Recommendation. *International Journal of Adult Education and Technology*, **2023**, 14(1), 1-16.
8. Manikandan, N. K.; Kavitha, M. Enhancing Content Recommendation System using Semantic Awareness with Flamingo Search Optimization and Bi-LSTM. In proceedings of the IEEE International Conference on Sustainable Computing and Smart Systems (June 2023)
9. Mishra, P.; Jain, V. Course Recommendation System using Content-based Filtering. In proceedings of the IEEE 7th International Conference on Trends in Electronics and Informatics (11 April 2023)
10. Noorian Avval, A. A.; Harounabadi, A. A hybrid recommender system using topic modeling and prefixspan algorithm in social media. *Complex & Intelligent Systems*, **2023,** 9(4), 4457-4482.
11. Sharma, E.; Jos, J. Recommendation System using Clustering and Comparing Clustering and Topic Modelling Techniques. In proceedings of the 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (28 April 2022)
12. Krinidis, S.; Chatzis, V. A robust fuzzy local information C-means clustering algorithm. In proceedings of the IEEE transactions on image processing (19 January 2010)
13. Kowsari, K.; Heidarysafa, M.; Brown, D. E.; Meimandi, K. J.; Barnes, L. E. Rmdl: Random multimodel deep learning for classification. In Proceedings of the 2nd international conference on information system and data mining (9 April 2018)
14. Kaveh, A.; Zaerreza, A. Shuffled shepherd optimization method: a new meta-heuristic algorithm. *Engineering Computations*, **2020**, 37(7), 2357-2389.
15. MiarNaeimi, F.; Azizyan, G.; Rashki, M. Horse herd optimization algorithm: A nature-inspired algorithm for high-dimensional optimization problems. *Knowledge-Based Systems*, **2021,** 213, 106711.
16. Choudhary, C.; Singh, I.; Kumar, M. SARWAS: Deep ensemble learning techniques for sentiment-based recommendation system*. Expert Systems with Applications*, **2023**, 216, 119420.
17. Raja, J. G. J. S.; Juliet, S. Deep Learning-based Sentiment Analysis of Trip Advisor Reviews. In proceedings of the 2nd International Conference on Applied Artificial Intelligence and Computing (4 May 2023)
18. Deepak, G.; Trivedi, I. A Hybridized Deep Learning Strategy for Course Recommendation. *International Journal of Adult Education and Technology*, **2023**, 14(1), 1-16.
19. Berkani, L.; Zaouidi, M.; Brahimi, R. Sentiment deep learning algorithm for multi-criteria recommendation. In proceedings of the 1st IEEE International Conference on Big Data, IoT, Web Intelligence and Applications (11 December 2022)
20. Sharma, E.; Jos, J. Recommendation System using Clustering and Comparing Clustering and Topic Modelling Techniques. In proceedings of the 2nd IEEE International Conference on Advance Computing and Innovative Technologies in Engineering (28 April 2022)
21. Karabila, I.; Darraz, N.; El-Ansari, A.; Alami, N.; El Mallahi, M. Enhancing collaborative filtering-based recommender system using sentiment analysis. *Future Internet*, **2023**, 15(7), 235.

22. Lakkaraju, K.; Srivastava, B.; Valtorta, M. Rating sentiment analysis systems for bias through a causal lens., In proceedings of the IEEE Transactions on Technology and Society (11 March 2024)
23. Guo, S.; Zhang, S.; Sun, W.; Ren, P., Chen; Z.; Ren, Z. Towards explainable conversational recommender systems. In Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (19 July 2023)
24. Karabila, I.; Darraz, N.; El-Ansari, A.; Alami, N.; El Mallahi, M. Enhancing collaborative filtering-based recommender system using sentiment analysis. *Future Internet*, **2023**, 15(7), 235.
25. Vu-Thi, V.; Luong-The, D.; Hoang-Van, Q. An efficient privacy-preserving recommender system. In Proceedings of the 14th International Conference on Knowledge and Systems Engineering, IEEE (19 October 2022)
26. The Coursera Kaggle dataset. Available online: “https://www.kaggle.com/” (accessed on 06 August, 2024).
27. Severyn, A.; Moschitti, A. Twitter Sentiment Analysis with Deep Convolutional Neural Networks. In Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM (9 August 2015)
28. Tang, D.; Qin, B.; Liu, T. Document Modeling with Gated Recurrent Neural Network for Sentiment Classification. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (September 2015)
29. Qu, C.; Ji, F.; Qiu; M.; Yang, L.; Min, Z.; Chen, H.; Huang, J; Croft, W.B. Learning to Selectively Transfer: Reinforced Transfer Learning for Deep Text Matching. In Proceedings of the 12th ACM international conference on web search and data mining (30 January 2019)
30. Bottou, L. Large-scale machine learning with stochastic gradient descent. In Proceedings of COMPSTAT'2010: 19th International Conference on Computational Statistics, Paris, France, (August 2010)
31. Kingma, D. P.; Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412*. **2014**, 6980.
32. Ruder, S. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609*. **2016**, 4747.
33. Zhang, S.; Yao, L.; Sun, A.; Tay, Y. Deep Learning Based Recommender System: A Survey and New Perspectives. *ACM Computing Surveys (CSUR)*, **2019**, 52(1), 1-38.
34. Ouhbi, B.; Frikh, B.; Zemmouri, E.; Abbad. Deep learning-based recommender systems. In proceedings of the IEEE 5th international congress on information science and technology (21 October 2018)
35. Mu, R. A survey of recommender systems based on deep learning. *Ieee Access*, **2018**, 6, 69009-69022.
36. He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X.; Chua, T.-S. Neural Collaborative Filtering. In Proceedings of the 26th International Conference on World Wide Web (3 April 2017)
37. Wang, H.; Zhang, F.; Zhao, M.; Li, W.; Xie, X.; Guo, M. Multi-Task Feature Learning for Knowledge Graph Enhanced Recommendation. In Proceedings of the ACM 2019 World Wide Web Conference (13 May 2019)

38. Roy, D.; Dutta, M. Optimal hierarchical attention network-based sentiment analysis for movie recommendation. *Social Network Analysis and Mining*, **2022**, 12(1), 138.
39. Kumar, V.; Thakur, G. S. M. Advanced Applications of Deep Learning in Big Data Analytics: An Overview. *Big Data Mining and Analytics*, **2020**, 3(1), 11-19.
40. Singh, M. Scalability and sparsity issues in recommender datasets: a survey. *Knowledge and Information Systems*, **2020**, 62(1), 1-43.
41. Roy, D.; Dutta, M. An improved cat swarm search-based deep ensemble learning model for group recommender systems. *Journal of Information & Knowledge Management*, **2022**, 21(03), 2250032.
42. Shambour, Q. A deep learning-based algorithm for multi-criteria recommender systems. *Knowledge-based systems*, **2021**, 211, 106545.
43. Alhijawi, B.; Kilani, Y. A collaborative filtering recommender system using genetic algorithm. *Information Processing & Management*, **2020**, 57(6), 102310.
44. Da'u, A.; Salim, N. Sentiment-aware deep recommender system with neural attention networks. *IEEE Access*, **2019**, 7, 45472-45484