mathematics

Article

Taylor-ChOA: Taylor-Chimp Optimized Random Multimodal Deep Learning-Based Sentiment Classification Model for Course Recommendation

Santosh Kumar Banbhrani [1,*], Bo Xu [1], Hongfei Lin [1] and Dileep Kumar Sajnani [2]

[1] School of Computer Science and Technology, Dalian University of Technology, Ganjingzi District, Dalian 116024, China; xubo@dlut.edu.cn (B.X.); hflin@dlut.edu.cn (H.L.)

[2] School of Computer Science and Engineering, Southeast University, Nanjing 210096, China; sajnani.dileep@gmail.com

* Correspondence: banbhrani@gmail.com

Abstract: Course recommendation is a key for achievement in a student's academic path. However, it is challenging to appropriately select course content among numerous online education resources, due to the differences in users' knowledge structures. Therefore, this paper develops a novel sentiment classification approach for recommending the courses using Taylor-chimp Optimization Algorithm enabled Random Multimodal Deep Learning (Taylor ChOA-based RMDL). Here, the proposed Taylor ChOA is newly devised by the combination of the Taylor concept and Chimp Optimization Algorithm (ChOA). Initially, course review is done to find the optimal course, and thereafter feature extraction is performed for extracting the various significant features needed for further processing. Finally, sentiment classification is done using RMDL, which is trained by the proposed optimization algorithm, named ChOA. Thus, the positively reviewed courses are obtained from the classified sentiments for improving the course recommendation procedure. Extensive experiments are conducted using the E-Khool dataset and Coursera course dataset. Empirical results demonstrate that Taylor ChOA-based RMDL model significantly outperforms state-of-the-art methods for course recommendation tasks.

Keywords: chimp optimization algorithm; course recommendation; E-learning; long short-term memory; random multimodal deep learning; sentiment classification

MSC: 68T50

Citation: Banbhrani, S.K.; Xu, B.; Lin, H.; Sajnani, D.K. Taylor-ChOA: Taylor-Chimp Optimized Random Multimodal Deep Learning-Based Sentiment Classification Model for Course Recommendation. *Mathematics* **2022**, *10*, 1354. https://doi.org/10.3390/math10091354

Academic Editors: Florentina Hristea and Cornelia Caragea

Received: 26 December 2021
Accepted: 11 April 2022
Published: 19 April 2022

1. Introduction

E-learning, termed as learning experiences or instructional content-enabled or delivered by electronic technology, particularly standalone computers and computer networks, is one of the foremost modernization that is gradually diffusing into community settings. In addition, web-driven intelligent E-learning environments (WILE) have gained significant attraction across the world, as they bear the power to enhance the superiority of E-learning services and applications. WILE can resolve the major limitation of E-learning methodologies by promoting adapted learning experiences, personalized to the specific individuality of every learner [1]. A course review process should find the quality of individual courses and establish areas in each course and potentially more global areas for development. This process should concentrate on foundation aspects of learning, teaching, and assessment, namely the presence of suitable learning objectives; degree of learning-centered activities; assessment techniques consistent with course objectives; and learning goals. Moreover, the course review process should also inspect consistency in coordination and suitable course contents and policies [2]. Furthermore, sentiment analysis should be conducted to quantify the user emotions involved in the review data [3], and the sentiment assessment should evaluate the words utilized in reviews, which permits visitors to find whether past visitors had an overall bad or good understanding of the listing [4,5].

In E-learning, a course recommendation system recommends the optimal courses in which the students are participating [6,7]. Numerous studies have shown that users face difficulties when choosing a course on an online educational website [8] because of the massive quantity of data. The course selection process is time-consuming and challenging. The important information offered by a course recommendation system can include relevant resource information, such as users' interests and job opportunities. Hence, the course recommendation systems on online education websites must exploit a variety of resources to match the objectives, knowledge structure, and interests of individual users [9]. In addition, selecting a proper course of study is very significant, as the users' futures are dependent on these decisions. The course recommendation system is necessary to assist the student in selecting appropriate courses. It can give a solution to help the student receive the appropriate target outcomes. On the other hand, the process of selecting a personalized course can be highly challenging and intricate for the user [10]. Recently, the recommendation system has become popular in both industry and academia as it reduces the information overloading problem. In numerous applications, the recommendation systems make an effort to evaluate the targeted ratings of the user on unrated items and thereafter recommend the items with high predicted ratings in order to minimize the user attempts and accordingly improve user contentment. Furthermore, data sparsity is the most commonly known issue in recommendation systems in which users have ratings on a lesser number of items, which makes it more difficult to learn efficient recommendation models [11].

Review data can consider the preferences of the user on every rated product and its particular data and can be considered as a carrier of significant information, which will control the character of other prospective users [12]. Review text is considered to be more vital for effective item representation and user learning for recommendation systems. Earlier research work have revealed that incorporating user reviews into the optimization of recommendation systems can extensively enhance the rating performance by reducing data sparsity issues [13,14]. Currently, deep learning methods have gained attraction from various domains due to their significant performance when compared with different traditional approaches [15]. Motivated by the successful exploitation of deep neural networks on the natural language processing (NLP) process, recent work has been committed to modeling user reviews using deep learning methods. Moreover, the most widely used techniques concatenate item reviews and users reviews initially and then accomplish neural network-enabled techniques, such as convolutional neural networks (CNNs) [16], long short-term memory (LSTM), and gated recurrent units (GRUs) [17] for extracting the vector form of the concatenated reviews. Nevertheless, not all the reviews are valuable for the given recommendation task. To emphasize the key knowledge in the comments, a few models exploited attention mechanisms for capturing key information [18,19].

The objective of this research is to design a method, named TaylorChOA-based RMDL, for course recommendation in the E-Khool platform using sentiment analysis for finding positively reviewed courses. The proposed method involves various phases, such as matrix construction, course grouping, course matching, sentiment classification, and course recommendation. Here, the input review data are fed to the matrix construction phase to transform the review data into matrix form. After constructing the matrix, the courses are grouped using deep embedded clustering (DEC) and then the course matching is done using the RC coefficient. After course matching, relevant scholar retrieval and matching are done using the Bhattacharya coefficient to select the best course. In the sentiment classification phase, the significant features, like SentiWordNet-based statistical features, classification-specific features, such as all-caps, numerical words, punctuation marks, elongated words, and time frequency-inverse document frequency (TF-IDF) features are effectively extracted and then sentimental classification is performed using RMDL, which is trained by the proposed TaylorChOA method. The developed TaylorChOA is designed by the incorporation of the Taylor concept and ChOA.

An effective sentiment analysis-based course recommendation method is developed for recommending the positively reviewed courses to the scholars. The courses are grouped by using DEC and then utilized for the matching process, which is carried out using the RV coefficient. The Bhattacharya coefficient is employed for the relevant scholar retrieval and matching process to select the best course. Moreover, the RMDL is used for classifying the sentiments by determining the positively and negatively reviewed courses. The training practice of the RMDL is effectively done using the developed Taylor ChOA, which is the hybridization of the Taylor concept and ChOA.

The major contribution of the paper is a novel sentiment classification approach that is proposed for recommending the courses using Taylor ChOA-based RMDL. Here, the proposed Taylor ChOA is devised by the combination of the Taylor concept and ChOA.

The remainder of the paper is organized as follows. Section 2 describes the review of different course recommendation methods. In Section 3, we briefly introduce the architecture of the proposed framework. Systems implementation and evaluation are described in Section 4. Results and discussion are summarized in Section 5. Finally, Section 6 concludes the overall work and discusses future research studies.

2. Related Work

(a) Hierarchical Approach:

Chao Yang et al. [12] introduced a hierarchical attention network oriented towards crowd intelligence (HANCI) for addressing rating prediction problems. This method extracted more exact user choices and item latent features. Although valuable reviews and significant words provided a positive degree of explanation for the recommendation, this model failed to analyze the recommendation performance by explaining the method at the feature level. Hansi Zeng and Qingyao Ai [14] developed a hierarchical self-attentive convolution network (HSACN) for modeling reviews in recommendation systems. This model attained superior performance by extracting efficient item and user representations from reviews. However, this method suffers from computational complexity problems.

(b) Deep Learing Approach:

Qinglong Li and Jaekyeong Kim [10] introduced a novel deep learning-enabled course recommender system (DÉCOR) for sustainable improvement in education. This method reduced the information overloading problems. In addition, it achieved superior performance in feature information extraction. However, this method does not consider larger datasets to train the domain recommendation systems. Aminu Da'u et al. [20] modeled a multi channel deep convolutional neural network (MCNN) for recommendation systems. The model was more effective in using review text and hence achieved significant improvements. However, this method suffers from data redundancy problems. Chao Wang et al. [21] devised a demand-aware collaborative Bayesian variational network (DCBVN) for course recommendation. This method offered accurate and explainable recommendations. This model was more robust against sparse and cold start problems. However, this method had higher time complexity.

(c) Query-based Approach:

Muhammad Sajid Rafiq et al. [22] introduced a query optimization method for course recommendation. This model improved the categorization of action verbs to a more precise level. However, the accuracy of online query optimization and course recommendation was not improved using this technique.

(d) Other Approaches:

Yi Bai et al. [19] devised a joint summarization and pre-trained recommendation (JSPTRec) for the recommendation based on reviews. This method learned improved semantic representations of reviews for items and users. However, the accuracy of rate prediction needed to be improved. Mohd Suffian Sulaiman et al. [23] designed a fuzzy logic approach for recommending the optimal courses for learners. This method significantly helped the students choose their course based on interest and skill. However, the sentiment analysis of user reviews was not considered for effective performance.

3. Proposed Method

The overall architecture of TaylorChOA-based RMDL method for sentiment analysis-based course recommendation, illustrated in Figure 1, contains several components. The detail of each component is presented next.

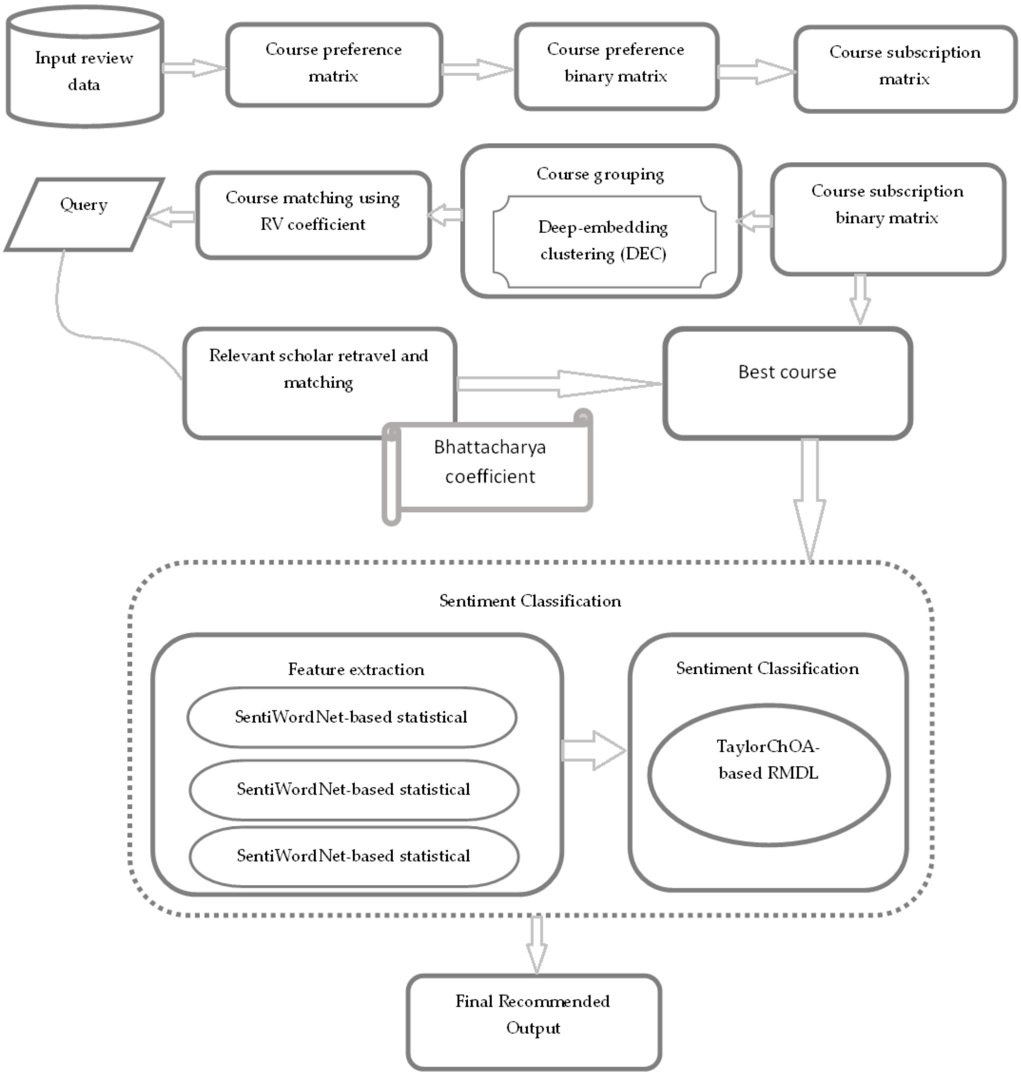

Input review data
Course preference matrix
Course preference binary matrix
Course subscription matrix
Query
Course matching using RV coefficient
Course grouping
Deep-embedding clustering (DEC)
Course subscription binary matrix
Relevant scholar retravel and matching
Best course
Bhattacharya coefficient
Sentiment Classification
Feature extraction
SentiWordNet-based statistical
SentiWordNet-based statistical
SentiWordNet-based statistical
Sentiment Classification
TaylorChOA-based RMDL
Final Recommended Output

Figure 1. An illustration of TaylorChOA-based RMDL for sentiment analysis-based course recommendation.

Initially, the input review data are presented to the matrix construction phase to construct the matrix based on learners' preferences. Thereafter, the constructed matrix is presented to the course grouping phase so that similar courses are grouped in one group, whereas different courses are grouped in another group using DEC [24]. When the query arrives, course matching is performed using the RV coefficient to identify the best course groups from overall course groups. After finding the best course group, relevant scholar retrieval and matching are performed between the user query and best course group using the Bhattacharya coefficient to find the best course. Once course review is performed, sentimental classification is carried out by extracting the significant features, such as SentiWordNet-based statistical features, classification-specific features, and TF-IDF features. Finally, sentiment classification is done using RMDL [25] that is trained by the developed TaylorChOA, which is the integration of the Taylor concept [26] and ChOA [27]. Finally, the positively recommended reviews are provided to the users. Figure 1 portrays a schematic representation of the sentiment analysis-based course recommendation model using the proposed TaylorChOA-based RMDL.

3.1. Acquisition of Input Data

The input dataset consists of a set of scholars lists and course lists.

Let the scholar's list be given as

$$D_s = \{S_i\} 1 < i \leq n \tag{1}$$

where n represents the total number of scholars, and S_i denotes i_{th} scholar. Each scholar learns a specific course. Let the course list be expressed as

$$D_c = \{C_j\} 1 < j \leq m \tag{2}$$

where m signifies the overall courses.

3.2. Matrix Construction

The input data are transformed to matrix form to make the course recommendation process simpler and more effective.

Course preference matrix: The input data D_s are acquired from the dataset and presented to the course preference matrix U_i. Each course has a specific ID that is denoted as service ID, and the Scholar ID who searched for the specific course is represented in the visitor preference matrix. The list of courses searched by scholars is given by

$$U_i = \left\{C_1^i, C_2^i, \ldots, C_l^i, \ldots, C_k^i\right\} \tag{3}$$

where C_1^i represents the l_{th} course preferred by scholar i, U_i indicates the course preferred by scholar i, and the total number of preferred courses is specified as k.

Course preference binary matrix: Once the course preference matrix U_i is generated, the course preference binary matrix B^{U_i} is performed based on the courses preferred, which is denoted as 0 and 1. For each course, the corresponding binary values of every course are given in the binary sequence. If a scholar preferred a course, then it is represented as 1, otherwise it is represented as 0. The course preference binary matrix is expressed as

$$B^{U_i} = \begin{cases} 1 & C_l^i \in C_j \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where B^{U_i} represents the course preference binary matrix for the scholar i.

Course subscription matrix: The course subscription binary matrix UL_j specifies the scholar who searches for a particular course. Thus, the courses searched by scholar are given as

$$UL_j = \left\{s_1^j, s_2^j, \ldots, s_p^j, \ldots, s_x^j\right\} \tag{5}$$

where, s_p^j indicates the j^{th} course searched by p^{th} scholar, x denotes the total number of scholars.

Course subscription binary matrix: After generating the course subscription matrix UL_j, the course subscription binary matrix B^{UL_j} is constructed based on courses subscribed, which is represented either as 0 or 1. For each course, the corresponding binary values for the subscribed course are given in the binary sequence. If the scholar searched for a course, it is denoted as 1, otherwise it is denoted as 0. The course subscription binary matrix is given as

$$B^{UL_j} = \begin{cases} 1 & S_p^j \in S_i \\ 0 & \text{Otherwise} \end{cases} \tag{6}$$

3.3. Course Grouping Using DEC Algorithm

The course grouping is performed using the DEC algorithm [24] for finding the best course groups. The DEC algorithm simultaneously learns the cluster assignments and feature representations by deep neural networks. This algorithm optimizes the clustering objective by understanding the mapping features from the data space to a low-dimensional space. It comprises two different phases, namely parameter initialization and clustering optimization, in which the auxiliary target distribution is computed and the Kullback–Leibler (KL) divergence is minimized.

The optimization of parameter or clustering optimization is illustrated by assuming a primary estimate of θ and $\{\ell_j\}_j^k = 1$.

Clustering with KL convergence: By considering an initial estimate of cluster centroids $\{\ell_j\}_j^k = 1$ and non-linear mapping f_θ, an unsupervised algorithm with two steps is devised for improving the process of clustering. In the initial phase, soft assignment is measured among the cluster centroids and embedded points. In the second phase, deep mapping f_θ is updated and the cluster centroids are refined based on the present high confidence assignments in terms of the auxiliary target distribution. This procedure is iteratively performed until the convergence condition is satisfied.

Soft assignment: Here, the student's t-distribution is used as a kernel for measuring the similarity among the centroid ℓ_j and embedded point S_i.

$$H_{ij} = \frac{\left(1 + \|s_i - \ell_j\|^2/\alpha\right)^{\frac{\alpha+1}{2}}}{\sum\left(1 + \left\|s_i - \ell_{j'}\right\|^2/\alpha\right)^{-\frac{\alpha+1}{2}}} \tag{7}$$

where $\ell_j = f_\theta(y_i) \in S$ corresponds to after the process of embedding, the degree of freedom is represented as α, and H_{ij} denotes the probability of sample i to cluster j.

KL divergence optimization: KL divergence optimization is designed for refining the clusters iteratively by understanding their assignments with higher confidence using the auxiliary target function. It computes the loss of convergence a_i among the auxiliary distribution and soft assignment b_i.

$$L = KL(P\|Q) = \sum_i \sum_j a_{ij} \log \frac{a_{ij}}{b_{ij}} \tag{8}$$

Furthermore, the computation is done by initially raising to the second power and thereafter normalizing the outcome by frequency per cluster.

$$a_{ij} = \frac{b_{ij}^2/f_j}{\sum_{j'} b_{ij'}^2/f_j{}'} \tag{9}$$

where $f_j = \sum_j b_{ij}$ represents the frequency of soft cluster. Hence, the DEC algorithm effectively improves low confidence prediction results.

The process of course grouping is done to group similar courses into their groups. The course grouping is performed among the scholars and courses. Let the course group obtained by deep embedded clustering be expressed as

$$G = \{G_1, G_2, \ldots, G_n\} \tag{10}$$

where n denotes the total number of groups. Thus, the output obtained by the course grouping in finding and grouping the course is denoted as G.

3.4. Course Matching Using RV Coefficient

The course matching is done using the RV coefficient where the user query is transformed to a binary query so that the matching operation is done to retrieve the best groups. The steps are elucidated below.

User query: When the user query arrives, the sequence of queries is given as

$$Q_z = \{q_1, q_2, \ldots, q_d, \ldots, q_r\} \tag{11}$$

where q_d specifies the total number of courses in query d and r represents the total number of queries.

Binary query sequence: The sequence of queries is transformed to binary query sequence formulated as

$$B^{Q_=} = \begin{cases} 1; & q_d \in C_j \\ 0; & \text{Otherwise} \end{cases} \tag{12}$$

where q_d denotes the number of courses in query d and B^{Q_z} represents the binary query sequence.

Course matching using RV coefficient: The course grouping is done using the RV coefficient by considering the course grouped sequence G and binary query sequence B^{Q_z}. Moreover, the RV coefficient is defined as the multivariate rationalization of the squared Pearson correlation coefficient because the RV coefficient considers the values within the range of 0 and 1. It measures the proximity of two sets of points characterized in a matrix form. The RV coefficient equation is given as follows:

$$RV\left(B^{Q_z}, G\right) = \frac{\text{Cov}\left(B^{Q_z}, G\right)}{\sqrt{\text{Var}\left(B^{Q_z}\right)\text{Var}(G)}} \tag{13}$$

where RV indicates the RV coefficient between $\left(B^{Q_z}, G\right)$, B^{Q_z} denotes the binary sequence, G specifies the grouped course, Cov represents the co-variance of $\left(B^{Q_z}, G\right)$, and Var specifies the variance of $\left(B^{Q_z}, G\right)$.

3.5. Relevant Scholar Retrieval

After performing the course matching, the relevant scholar retrieval is performed for identifying the best course group in a binary form. The scholar ID is identified based on the best group binary value, and the best course is preferred for scholars. Here, the list of courses is examined in terms of the scholars who are in the best groups.

Best course group: The best course group R_c for the relevant scholar retrieval is expressed as

$$R_C = \left\{r_1^i, r_2^i, \ldots, r_y^i, \ldots, r_w^i\right\} \tag{14}$$

where w represents the total number of best courses, and r_y^i denotes the best course retrieved by the scholar i.

Binary best course group: For each best course group, the corresponding binary values for the retrieved best course are given in a binary sequence. If the best course is retrieved by the scholar, it is indicated as 1, otherwise it is denoted as 0.

$$B^{R_C} = \begin{cases} 1; & r_y^i \in C_j \\ 0; & \text{Otherwise} \end{cases} \tag{15}$$

Matching query and best course group using Bhattacharya coefficient: Once the scholar retrieved the best course, the binary query sequence B^{Q_z} and the best course group B^{R_c} are compared using the Bhattacharya coefficient. The Bhattacharyya distance computes the similarity of two probability distributions, and the equation is expressed as

$$BC\left(B^{Q_=}, B^{R_C}\right) = \sum_{x \in X} \sqrt{P\left(B^{Q_E}\right) \cdot P\left(B^{R_C}\right)} \tag{16}$$

where BC indicates the Bhattacharya coefficient. Once the query and best group binary sequence are matched, the minimum value distance is chosen as the best course based on the Bhattacharya coefficient. The output of matching result is scholar preferred courses, and it is expressed as C_b, given as

$$C_b = \{C_1, C_2, \ldots, C_h\} \tag{17}$$

where C_h signifies courses preferred by a scholar that are the best courses. The best course C_b undergoes a sentimental classification process to verify whether the recommended course is good or bad. Algorithm 1 provides the Pseudo-code of course review framework.

Algorithm 1 Pseudo-code of Course Review Framework.

```
    Input: UserID: D_s , ItemID: $D_c$, Review: $R$, Query $Q_s$, Cluster size $C_s = 3$; Parameter $= U_i$ course preference matrix, $G$ best-clustered course group, $R_C$ relevant scholar retrieved, $n$ courses in optimal clustered group, $B^U$ course preference binary matrix, $n$ number of scholars, $m$ number of courses, $k$ is the total number of preferred course.
    Output: Best course $C_b$
Begin
Read Input $(D_{(s)}, D_{(c)}, R)$; $B^{(U_{(i)})}, B^{(UL_{(j)})} = U_{(i)}(D_{(s)}, D_{(c)})$
$G = DEC(B^{(UL_{(j)})}$, clustersize = 3)  Find $G$
$G$ = course Matching phase $(Q_z, G)$
Compute $R_C$ = Relevant visitor phase $\left(\mathrm{n}, \mathrm{B}^{\mathrm{U}^{(\mathrm{i})}}\right)$;
$C_b$ = Matched visitor phase $(Q_z, R_C)$ //Course preference matrix phase $B^{U_i} = (D_s, D_c)$
if scholar search the course; Print 1
else Print 0
$B^{UL_j} = (D_s, D_c)$
if ($m$ course is visited by the scholar) Print 1
else Print 0
$Q_z$ generation based on $B^{UL_j}$ //Course matching phase
$RV.grp = []$
for $j = 1$ to $G$ $Sum_{RV_{val}} = 0$
For    j = 1 to $len(h)$
$Sum_{RV_{val}} += $ RV coeff $(\mathrm{Q}_z,\ \mathrm{h})$
End for
RV.grp.app end($Sum_{RV_{val}}$)
End for
$G = \max(RV \cdot grp)$ //Relevant scholar phase
$R_C = []$
for j = 1 to $len$(h)
$C$ = got scholars who viewed the courses
$R_C$.append $\left(B^{U_i}(C)\right)$
End for
Return $R_C$
//Matched scholar phase $C_b = []$
for j = 1 to len
$(\mathrm{R}_C)$
$C_b$.append (Bhattacharya $(Q_z, R_C)$)
End for
Sort by min($\mathrm{C}_b$)
Return $C_b$
```

3.6. Sentiment Classification

The best course C_b is fed as an input to the sentiment classification phase to classify the sentiments in terms of the sentimental polarities of opinions. The classified sentiments may have either a positive score or a negative score.

Acquisition of significant features for sentiment classification: The significant features, such as SentiWordNet-based statistical features, classification-specific features, and TF-IDF features, are extracted from the best course C_b for improving the course recommendation process. The extracted features are elucidated below.

(a) SentiWordNet-based statistical features: SentiWordNet [28] groups the words into multiple sets of synonyms, called synsets. Every synset is associated with a polarity score, such as positive or negative. The scores take a value between 0 and 1, and their summation provides a value of 1 for every synset. By considering the scores provided, it is feasible to decide whether the estimation is positive or negative. The words present in the SentiWordNet database are based on the parts of speech attained from WordNet, and it utilizes a program to apply the scores to every word. The weight tuning of positive and negative score values can be expressed as

$$|\varphi^m(p), \varphi^m(n)| = h(w_m) \tag{18}$$

where $\varphi^m(p)$ represents the positive score, $\varphi^m(n)$ denotes the negative score, and h specifies the SentiWordNet function. However, the SentiWordNet feature is denoted as F_n. With the SentiWordNet score, statistical features, such as mean and variance, are computed using the expressions given below.

(i) Mean: The mean value is computed by taking the average of SentiWordNet score for every word from the review, given as

$$\mu = \frac{1}{|U(x_n)|} \times \sum_{n=1}^{|U(x_n)|} U(x_n) \tag{19}$$

where n represents the overall words, $U(x_n)$ signifies the SentiWordNet score of each review, and $|U(x_n)|$ represents the overall scores obtained from the word.

(ii) Variance: The variance σ is computed based on the value of the mean, given as

$$\sigma = \frac{\sum_{n=1}^{|U(x_n)|} |x_n - \mu|}{U(x_n)} \tag{20}$$

where μ signifies the mean value. Thus, the sentiwordNet-based feature considers the positive and negative scores of each word in the review, and from that, the statistical features, like mean and variance, are computed.

(b) Classification-specific features: The various classification specific features, such as capitalized words, numerical words, punctuation marks, and elongated words are explained below.

(i) All caps: The feature f_1 specifies the all-caps feature, which represents the overall capitalized words in a review, expressed as

$$f_1 = \sum_{m=1}^{b} w_C^m \tag{21}$$

where w_C^m indicates the total number of words with upper case letters. It considers a value 0 or 1 concerning the state that relies on the absence or presence of capitalized words as formulated below:

$$w_C^m = \begin{cases} 1; & \text{if capsword} \\ 0; & \text{otherwise} \end{cases} \tag{22}$$

Here, the feature f_1 is in the dimension of $[10{,}000 \times 1]$.

(ii) Number of numerical words: The number of text characters or numerical digits used to show numerals are represented as f_2 with the dimension $[10{,}000 \times 1]$.

(iii) Punctuation: The punctuation feature f_3 may be an apostrophe, dot, or exclamation mark present in a review:

$$f_3 = \sum_{m=1}^{b} S_p^m \tag{23}$$

where S_p^m represents the overall punctuation present in the m_{th} review. Here, S_p is given a value of 1 for the punctuation that occurred in the review and 0 for other cases. Moreover, the feature f_3 has the dimension of $[10{,}000 \times 1]$.

(iv) Elongated words: The feature f_4 represents the elongated words that have a character repeated more than two times in a review and is given as

$$f_4 = \sum_{m=1}^{b} w_E^m \tag{24}$$

where w_E^m specifies the overall hashtags present in the mth review. The term is given with a value of 0 for every elongated word in the review and 1 for the nonexistence of an elongated word. Furthermore, the elongated word feature f_4 holds the size of $[10{,}000 \times 1]$.

The classification specific features are signified as F_2 by considering the seven extracted features and is given as

$$F_2 = \{f_1, f_2, f_3, f_4\} \tag{25}$$

where f_1 denotes the all-caps feature, f_2 signifies the numerical word feature, f_3 specifies the punctuation feature, and f_4 indicates the elongated word feature.

(c) TF-IDF: TF-IDF [29] is used to create a composite weight for every term in each of the review data. TF measures how frequently a term occurs in review data, whereas IDF measures how significant a term is. The TF-IDF score is computed as

$$F_4 = C\left(\frac{\log(1+\phi_1)}{\log(\phi_2)}\right) \tag{26}$$

where C specifies the total number of review data, term frequency is denoted as ϕ_1, ϕ_2 represents the inverse document frequency, and F_3 implies the TF-IDF feature with dimension $[1 \times 50]$.

Furthermore, the features extracted are incorporated together to form a feature vector F for reducing the complexity in classifying the sentiments, which is expressed as

$$F = \{F_1, F_2, F_3\} \tag{27}$$

where F_1 signifies the SentiWordNet-based statistical feature, F_2 represents the classification specific features, F_3 implies the TF-IDF features, and F implies the feature vector with dimension $[10{,}000 \times 834]$.

3.7. Sentiment Classification Using Proposed TaylorChOA-Based RMDL

Here, the feature vector F is employed for classifying the sentiments effectively. The classification of sentiments is carried out using the proposed TaylorChOA-based RMDL. The RMDL [25] is trained with the proposed TaylorChOA algorithm, which is developed by combining the Taylor concept [26] and ChOA [27]. Thus, effective course recommendation is achieved by offering suitable courses for the learners. The architecture and training procedure of RMDL are explained below.

(a) Architecture of RMDL: RMDL [25] is a robust method that comprises three basic deep learning models, namely deep neural networks (DNN), recurrent neural networks (RNN), and a convolutional neural network (CNN) model. The structure of RMDL is presented in Figure 2.

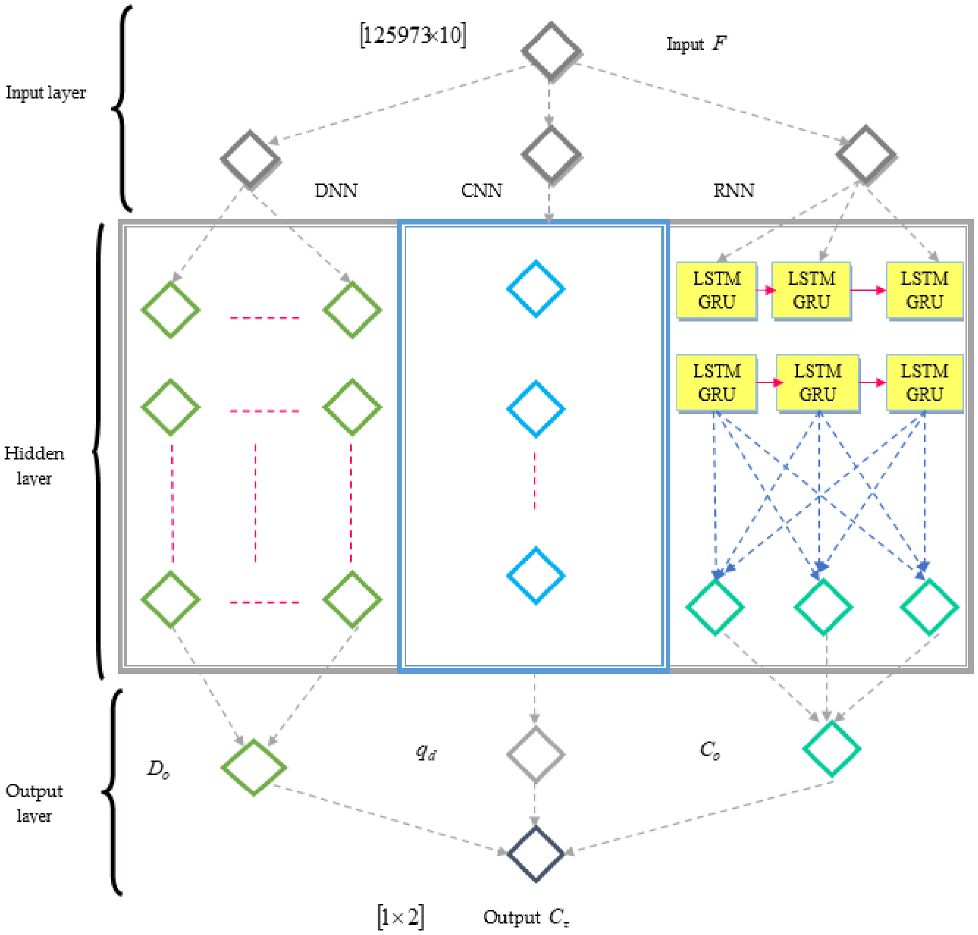


Figure 2. An illustration of random multimodel deep learning for sentiment analysis-based course recommendation.

(i) DNN: DNN architecture is designed with multi-classes where every learning model is generated at random. Here, the overall layer and its nodes are randomly assigned. Moreover, this model utilizes a standard back-propagation algorithm using activation functions. The output layer has a softmax function to perform the classification and is given as

$$f(x) = \frac{1}{1+g^{-x}} \in (0,1) \tag{28}$$

$$f(x) = \max(0, x) \tag{29}$$

The output of DNN is denoted as D_o.

(ii) RNN: RNN assigns additional weights to the sequence of data points. The information about the preceding nodes is considered in a very sophisticated manner to perform the effective semantic assessment of the dataset structure.

$$x_y = F_f(x_{y-1}, h_y, \theta) \tag{30}$$

$$x_y = U_{rec}\kappa(x_{y-1}) + U_{in}h_y + A \tag{31}$$

Here, x_y signifies the state at the time y, and h_y denotes the input at phase y. In addition, the recurrent matrix weight and input weight are represented as U_{rec} and U_{in}, the bias is represented as A, and κ indicates the element-wise operator.

Long short-term memory (LSTM): LSTM is a class of RNN that is used to maintain long-term relevancy in an improved manner. This LSTM network effectively addresses the vanishing gradient issue. LSTM consists of a chain-like structure and utilizes multiple gates for handling huge amounts of data. The step-by-step procedure of LSTM cell is expressed as follows:

$$F_d = \Re(w_F[p_d, q_{d-1}] + H_F) \tag{32}$$

$$\widetilde{C}_d = \tan q(w_C[p_d, q_{d-1}] + H_C) \tag{33}$$

$$r_d = \Re(w_r[p_d, q_{d-1}] + H_r) \tag{34}$$

$$J_d = F_d * \widetilde{C}_d + r_d J_{d-1} \tag{35}$$

$$M_d = \Re(w_M[p_d, q_{d-1}] + H_M) \tag{36}$$

$$q_d = M_d \tan y(J_d) \tag{37}$$

where F_d represents the input gate, $\widetilde{C}_d$ specifies the candidate memory cell, r_d denotes the forget gate activation, and J_d defines the new memory cell value. Here, M_d and q_d specify the output gate value.

Gated recurrent unit (GRU): GRU is a gating strategy for RNN that consists of two gates. Here, GRU does not have internal memory, and the step by step procedure for GRU cells is given as

$$N_d = \Re_l(w_N p_d + V_N q_{d-1} + H_z) \tag{38}$$

where n_d implies update gate vector of d, p_d denotes the input vector, the various parameters are termed as w, V, and H, and $\Re_l$ represent the activation parameter.

$$\widetilde{S}_d = \Re_l(w_S p_d + V_S q_{d-1} + H_S) \tag{39}$$

$$q_d = N_d \circ q_{d-1} + (1 - N_d) \circ \Re_l\left(w_q p_d + V_q(S_d \circ q_{d-1}) + H_d\right) \tag{40}$$

Here, q_d denotes the output vector, the reset gate vector is denoted as S_d, N_d indicates the update gate vector of d, and the hyperbolic tangent parameter is signified as $\Re_l$.

(iii) CNN: CNN is the final deep learning method that contributes to RMDL and is mainly accomplished for the classification process. In CNN, the convolution of an image tensor is done with a group of kernels with dimension $p \times p$. These types of convolutional layers are known as feature maps, and they are stacked to offer numerous input filters. To decrease the computational complexity, a pooling function is employed for reducing the output dimension from one layer to the next. Finally, the feature maps are flattened into one column in such a way that the last layer is fully connected. The output of CNN is expressed as C_0.

For these deep learning structures, the total number of nodes and layers are randomly generated. The random creation process is given by

$$T(t_{d1}, t_{d2}, t_{d3,\ldots,} t_{dt}) = \left\lfloor \frac{1}{2} + \frac{\left(\sum_{e=1}^{z} t_d\right) - \frac{1}{2}}{z} \right\rfloor \tag{41}$$

where z denotes the overall random models, t_{dz} specifies the output for a data point i in z, and this equation is utilized for classifying the sentiments, $k \in \{0, 1\}$. The output space uses majority vote for final $\hat{t}_d$, and the equation is expressed as

$$\hat{t}_d = \left[\hat{t}_{d1} \ldots \hat{t}_{de} \ldots \hat{t}_{dt}\right]^N \tag{42}$$

where $\hat{t}_d$ specifies the classification label of review or data point of $E_d \in \{a_d, b_d\}$ for e, and $\hat{t}_d$ is represented as follows:

$$\hat{t}_{d,z} = \arg\max_k \left[\text{softmax}\left(t^*_{d,z}\right)\right]^N \tag{43}$$

After training the RMDL model, the final classification is computed using a majority vote of DNN, CNN, and RNN models, which improve the accuracy and robustness of the results. The final result obtained from the RMDL is indicated as C_τ.

(b) Training of RMDL using the proposed TaylorChOA: The training procedure of RMDL [25] is performed using the developed optimization method, known as TaylorChOA. The developed TaylorChOA is designed by the incorporation of the Taylor concept and ChOA. ChOA [27] is motivated by the characteristics of chimps for hunting prey. It is mainly accomplished for solving the problems based on convergence speed by learning through the high dimensional neural network. In addition, the independent groups have different mechanisms for updating the parameters to explore the chimp with diverse competence in search space. The dynamic strategies effectively balance the global and local search problems. The Taylor concept [26] exploits the preliminary dataset and the standard form of the system for validating the Taylor series expansion in terms of a specific degree. The incorporation of the Taylor series with the ChOA shows the effectiveness of the developed scheme and minimized the computational complexity. The algorithmic procedure of the proposed TaylorChOA algorithm is illustrated below.

(i) Initialization: Let us consider the chimp population as $Z_i(i = 1, 2, \dots, m)$ in the solution space N, and the parameters are initialized as n, u, v, and r. Here, n specifies the non-linear factor, u implies the chaotic vector v, and r denotes the vectors.

(ii) Calculate fitness measure: The fitness measure is accomplished for calculating the optimal solution using the error function and is expressed as

$$\xi = \frac{1}{\ell} \sum_{\delta=1}^{\ell} [E_\tau - C_\tau]^2 \tag{44}$$

where ξ signifies fitness measure, E_τ specifies target output, ℓ indicates overall training samples, and the output of the RMDL model is denoted as C_τ.

(iii) Driving and chasing the prey: The prey is chased during the exploitation and exploration phases. The mathematical expression used for driving and chasing the prey is expressed as

$$Z(s+1) = Z_{\text{prey}}(s) - x \cdot y \tag{45}$$

where s represents the current iteration, x signifies the coefficient vector, Z_{prey} implies the vector of prey position, y indicates driving the prey, and the position vector of chimp is specified as Z. Here, y is expressed as

$$y = \left| r.Z_{\text{prey}}(s) - u.Z(s) \right| \tag{46}$$

Let us consider $Z_{\text{prey}}(s) > Z(s)$,

$$Z(s+1) = Z_{\text{prey}}(s) - x\left(r.Z_{\text{prey}}(s) - u.Z(s)\right) \tag{47}$$

$$Z(s+1) = Z_{\text{prey}}(s) - x.rZ_{\text{prey}}(s) + x.u.Z(s) \tag{48}$$

$$Z(s+1) = Z_{\text{prey}}(s)[1 - x.r] + x.u.Z(s) \tag{49}$$

By incorporating the Taylor concept [26] with the ChOA [27], the algorithmic performance is improved by minimizing the optimization problems. The standard equation of the Taylor concept [26] is expressed as

$$Z(s+1) = Z(s) + \frac{Z'(s)}{1!} + \frac{Z''(s)}{2!} \tag{50}$$

where

$$Z'(s) = \frac{Z(s) - Z(s-k)}{k} \tag{51}$$

$$Z''(s) = \frac{Z(s) - 2Z(s-k) + Z(s-2k)}{k^2} \tag{52}$$

Assume $s = 1$ and substitute $Z'(s), Z''(s)$ in Equation (50):

$$Z(s+1) = Z(s) + \frac{Z(s) - Z(s-1)}{1!} + \frac{Z(s) - 2Z(s-1) + Z(s-2)}{2!} \tag{53}$$

$$Z(s+1) = Z(s)\left(1 + 1 + \frac{1}{2}\right) - Z(s-1) - \frac{2Z(s-1)}{2} + \frac{Z(s-2)}{2} \tag{54}$$

By substituting Equation (56) in Equation (50), the equation becomes

$$Z(s+1) = \frac{5Z_{\text{mey}}\ (s)[1 - xs] + 4 \cdot x \cdot u \cdot Z(s-1) - x \cdot uZ(t-2)}{5 - 2x \cdot u} \tag{55}$$

where the coefficient vector is denoted as r, the position of a chimp at iteration $s - 1$ is specified as $Z(s-1)$, the position of a chimp at iteration $s - 2$ is specified as $Z(s-2)$, Z_{prey} indicates the vector of prey position, and u implies the chaotic value. Moreover, $x = 2 \cdot v \cdot w_1 - v$, and $r = 2 \cdot w_2$ where the value of v is reduced from 2.5 to 0 and w_1, w_2 denotes the random vector within the range $[0, 1]$.

(iv) Attacking strategy (exploitation phase): To mathematically formulate the attacking character of chimps, it is considered that the first attacker, driver, barrier, and chaser are informed regarding the position of potential prey. Thus, the four optimal solutions to update the position are given as

$$Z(s+1) = \frac{Z_1 + Z_2 + Z_3 + Z_4}{4} \tag{56}$$

(v) Prey attacking: In this prey attacking phase, the chimps attack the prey and end the hunting operation once the prey starts moving. To mathematically formulate the attacking behavior, the value must be decreased.

(vi) Searching for prey (exploration phase): The exploration process is performed based on the position of the attacker, chaser, barrier, and driver chimps. Moreover, chimps deviate to search for the prey and aggregate to chase the prey.

(vii) Social incentive: To acquire social meeting and related social motivation in the final phase, the chimps release their hunting potential. To model this process, there is a 50% chance to prefer between the normal position update strategy and chaotic model for updating the position of chimps during the optimization. The equation is represented as

$$Z_{\text{chimp}}\ (s+1) = \begin{cases} Z_{\text{prey}}\ (s) - x \cdot y & \text{if } \omega < 0.5 \\ \text{Chaticvalue} & \text{if } \omega > 0.5 \end{cases} \tag{57}$$

where, ω denotes the random number between $[0, 1]$.

(viii) Feasibility evaluation: The fitness value is calculated for each solution such that the best value of fitness is considered the best solution.

(ix) Termination: All the above-presented steps are iterated until the global optimal solution is achieved. Algorithm 2 provides the pseudo-code of the proposed TaylorChOA.

The developed TaylorChOA-based RMDL model achieved effective performance in recommending the positively reviewed courses to the scholars by classifying the positively and negatively reviewed courses.

Algorithm 2 Pseudo-code of proposed TaylorChOA algorithm.

```
    Input: $Z_i$
    Output: $Z_{chimp}(s+1)$
Initialize population
Initialize the parameters, like $v, u, x$, and $r$
Determine the position of each chimp
while $(s < \aleph)$; $\aleph -$ maximum iterations
for each chimp
Extract the group of chimps
Use the grouping mechanism to update $v, u$, and $r$
end for
for each search chimp
if $(\varpi < 0.5)$
if $(|x| < 1)$
Update position of search agent using Equation (56)
else if $(x > 1)$
Choose a random search agent
end if
else if $(\varpi > 0.5)$
Update position of search using the chaotic value
end if
end for
Update $v, u, x$, and $r$
$s = s + 1$
end while
Return the best solution
```

4. Systems Implementation and Evaluation

In this section, we first present the datasets, then details about the experimental setup, baseline benchmarks, and finally evaluation metrics are shown.

4.1. Description of Datasets

In order to evaluate our system, the E-Khool https://ekhool.com/ (accessed on 12 October 2021) and Coursera Course https://www.kaggle.com/siddharthm1698/coursera-course-dataset (accessed on 10 February 2022) datasets are adapted for sentiment classification based course recommendation.

The E-Khool dataset comprises 100,000 rows with 25 courses and 1000 learners. This dataset includes various attributes, such as learner ID, course ID, subscription date, ratings (1 to 5), and review.

The Coursera Course dataset was generated during a hackathon for project purposes. It contains 6 columns and 890 course data points. The columns are course-title, course-organization, course-certificate-type, course-rating, course-difficulty, and course-students-enrolled.

4.2. Experimental Setup

The method we proposed is implemented in Python programming language; our networks are trained on NVIDIA GTX 1080 in a 64-bit computer with Intel(R) Core(TM) i7-6700 CPU @3.4GHz, 16 GB RAM, and Ubuntu 16.04 operating system.

4.3. Evaluation Metrics

The performance of the developed TaylorChOA-based RMDL is analyzed by considering the evaluation measures, like precision, recall, and F1-score.

Precision: This is the proportion of true positives to overall positives, and the precision measure is expressed as

$$\delta = \frac{A}{A+B} \tag{58}$$

where δ specifies the precision, A denotes the true positives, and B signifies the false positives.

Recall: Recall is a measure that defines the proportion of true positives to the summing up of false negatives and true positives, and the equation is given as

$$\omega = \frac{A}{A+E} \tag{59}$$

where the recall measure is signified as ω, and E symbolizes the false negatives.

F1-score: This is a statistical measure of the accuracy of a test or an individual based on the recall and precision, which is given as

$$F_m = 2 * \left(\frac{\delta * \omega}{\delta + \omega} \right) \tag{60}$$

where F_m denotes the F1-score.

4.4. Baseline Methods

In order to evaluate the effectiveness of the proposed framework, our method was compared with several existing algorithms, such as:

- HSACN [14]: The method was formulated to learn item and user representations from reviews.
- MCNN [20]: Multichannel Deep Convolutional Neural Network for Recommender Systems.
- Query Optimization [22]: The Query Optimization method for course recommendation model designed to improve the categorization of action verbs to a more precise level.
- DCBVN [21]: Demand-aware Collaborative Bayesian Variational Network for course recommendation.
- Proposed TaylorChOA-based RMDL: Proposed TaylorChOA-based RMDL model is developed for recommending the finest courses.

5. Results and Discussion

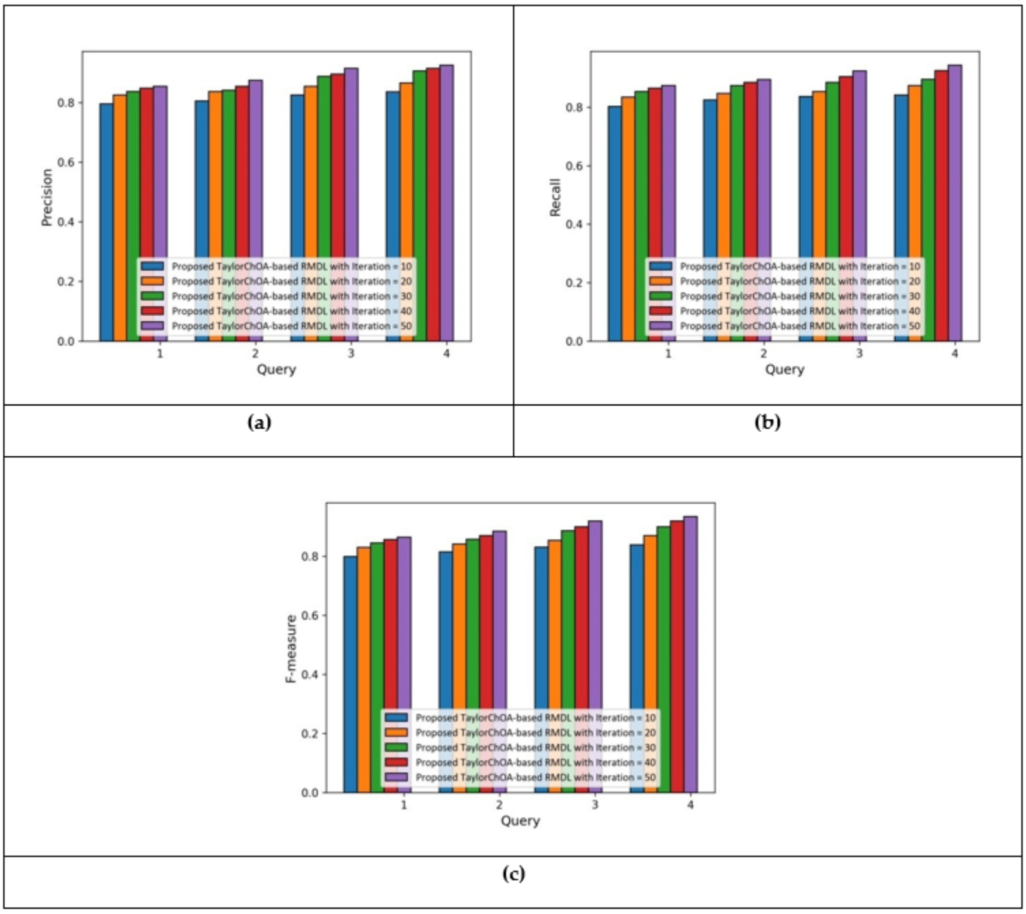
The performance results of our proposed model are presented in this section. The results are compared with previously introduced methods, which were tested on the same datasets.

5.1. Results Based on E-Khool Dataset, with Respect to Number of Iterations (10 to 50)

5.1.1. Performance Analysis Based on Cluster Size = 3

Figure 3 presents the performance analysis of the developed technique with iterations by varying the queries with cluster size =3. Figure 3a presents the assessment based on precision. For the number of query 1, the precision value measured by the developed TaylorChOA-based RMDL with iteration 10 is 0.795, iteration 20 is 0.825, iteration 30 is 0.836, iteration 40 is 0.847, and iteration 50 is 0.854. Figure 3b portrays the analysis using recall.

By considering the number of query 2, the value of recall computed by the developed TaylorChOA-based RMDL with iteration 10 is 0.825, iteration 20 is 0.847, iteration 30 is 0.874, iteration 40 is 0.885, and iteration 50 is 0.895. The analysis using F1-score is depicted in Figure 3c. When the number of a query is 3, the value of F1-score computed by the developed TaylorChOA-based RMDL with iteration 10 is 0.830, iteration 20 is 0.854, iteration 30 is 0.886, iteration 40 is 0.900, and iteration 50 is 0.919.



(a)

(b)

(c)

Figure 3. Performance analysis with cluster size = 3 using E-Khool dataset: (**a**) precision, (**b**) recall, and (**c**) F1-score.

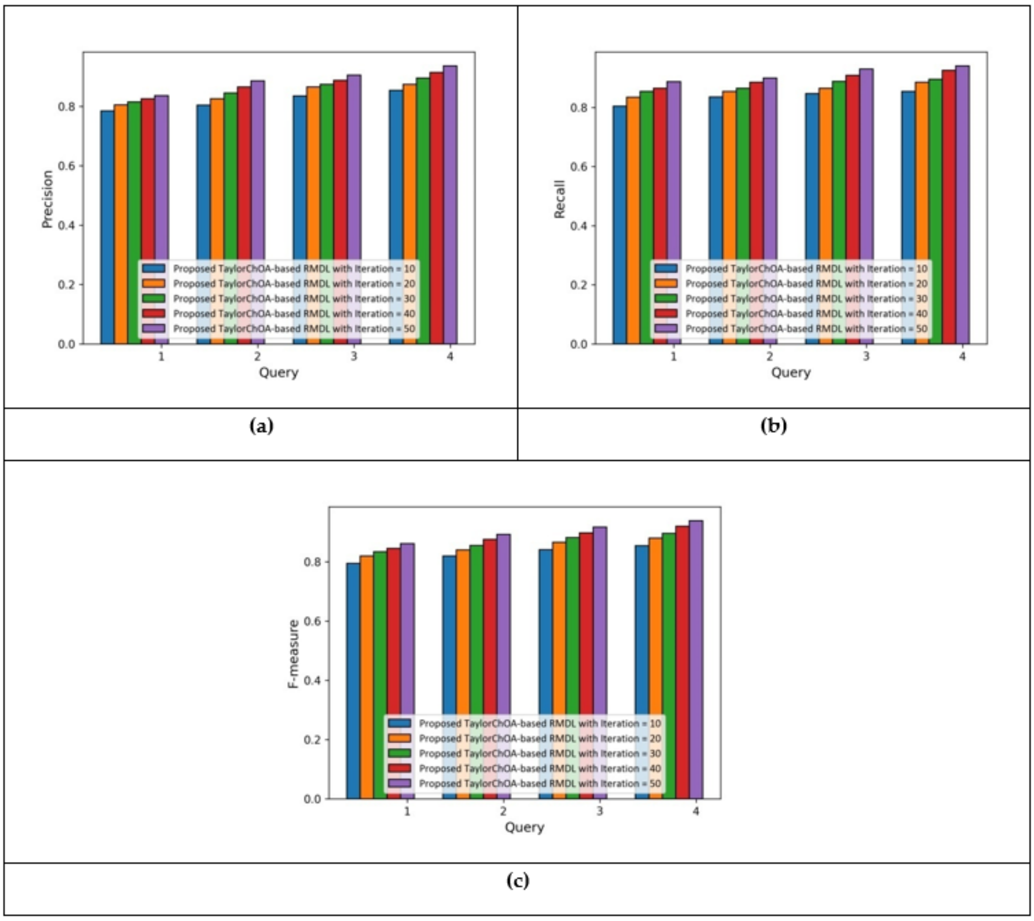
5.1.2. Performance Analysis Based on Cluster Size = 4

Figure 4 shows the performance assessment of the developed technique with iterations by varying the queries. Figure 4a presents the analysis based on precision. For the number of query 1, the precision value measured by the developed TaylorChOA-based RMDL with iteration 10, iteration 20, iteration 30, iteration 40, and iteration 50 is 0.784, 0.804, 0.814, 0.825, and 0.836, respectively. Figure 4b portrays the analysis using recall. By considering the number of query 2, the value of recall computed by the developed TaylorChOA-based RMDL with iteration 10 is 0.835, iteration 20 is 0.854, iteration 30 is 0.865, iteration 40 is 0.885, and iteration 50 is 0.899.

The analysis in terms of F1-score is shown in Figure 4c. When the number of a query is 3, the value of F1-score computed by the developed TaylorChOA-based RMDL with iteration 10 is 0.841, iteration 20 is 0.865, iteration 30 is 0.881, iteration 40 is 0.897, and iteration 50 is 0.917.

Comparison of existing methods and the proposed TaylorChOA-based RMDL using E-Khool dataset, in terms of precision, recall, and F1-score:

The comparative assessment of the developed technique is performed by varying the queries with the cluster size = 3 and cluster size = 4 in terms of the evaluation metrics.



(a)

(b)

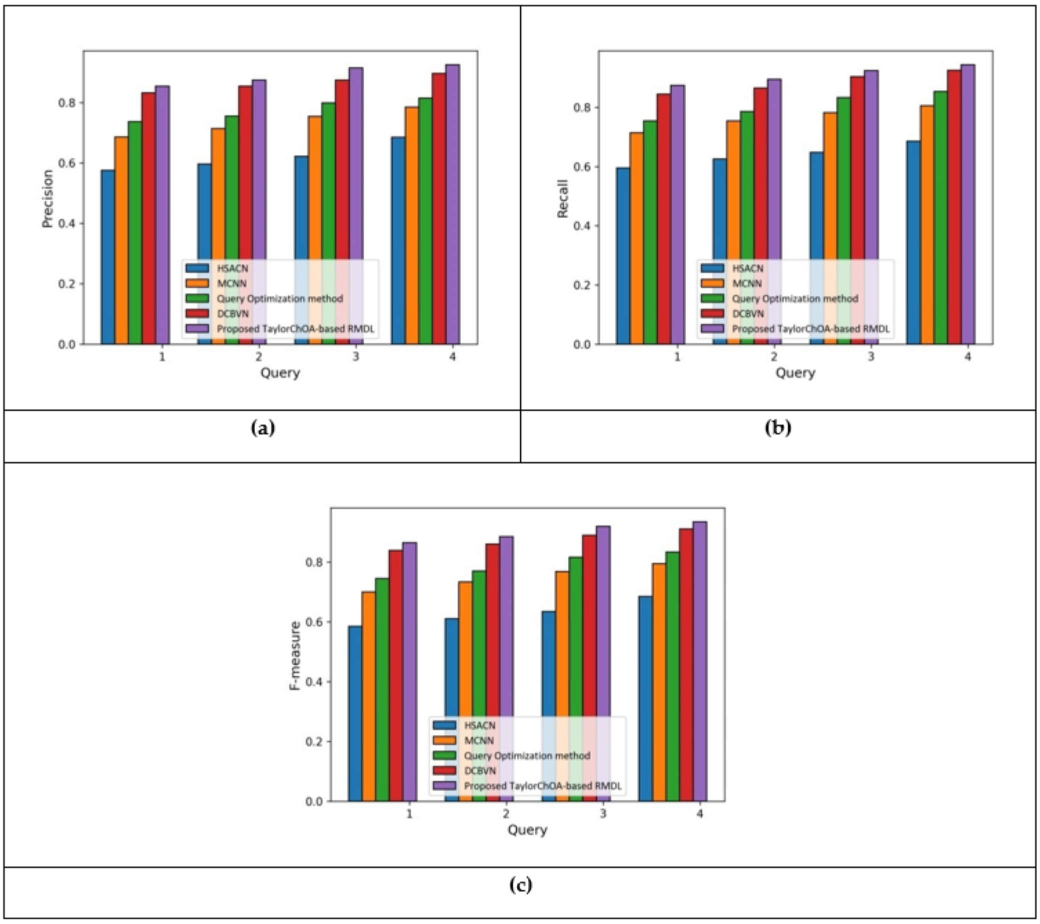
(c)

Figure 4. Performance analysis with cluster size = 4 using E-Khool dataset: (**a**) precision, (**b**) recall, and (**c**) F1-score.

5.1.3. Comparative Analysis Based on Cluster Size = 3 in terms of Precision, Recall, and F1-Score Using E-Khool Dataset

Figure 5 portrays the assessment with cluster size = 3 by varying the number of queries using the performance measures, such as precision, recall, and F1-score. Figure 5a) presents the analysis in terms of precision. When number of query is 1, the precision value measured by the developed TaylorChOA-based RMDL is 0.854, whereas the precision value measured by the existing methods, such as HSACN, MCNN, Query Optimization, and DCBVN is 0.575, 0.685, 0.736, and 0.832, respectively.

The analysis based on recall measure is portrayed in Figure 5b. By considering the number of query as 2, the developed TaylorChOA-based RMDL measured a recall value of 0.895, whereas the value of recall computed by the existing methods, such as HSACN, MCNN, Query Optimization, and DCBVN is 0.625, 0.754, 0.785, and 0.865, respectively. The assessment using F1-score is shown in Figure 5c. The F1-score value attained by the HSACN, MCNN, Query Optimization, DCBVN, and developed TaylorChOA-based RMDL is 0.634, 0.768, 0.815, 0.889, and 0.919, respectively, when considering the number of query as 3.



(a)

(b)

(c)

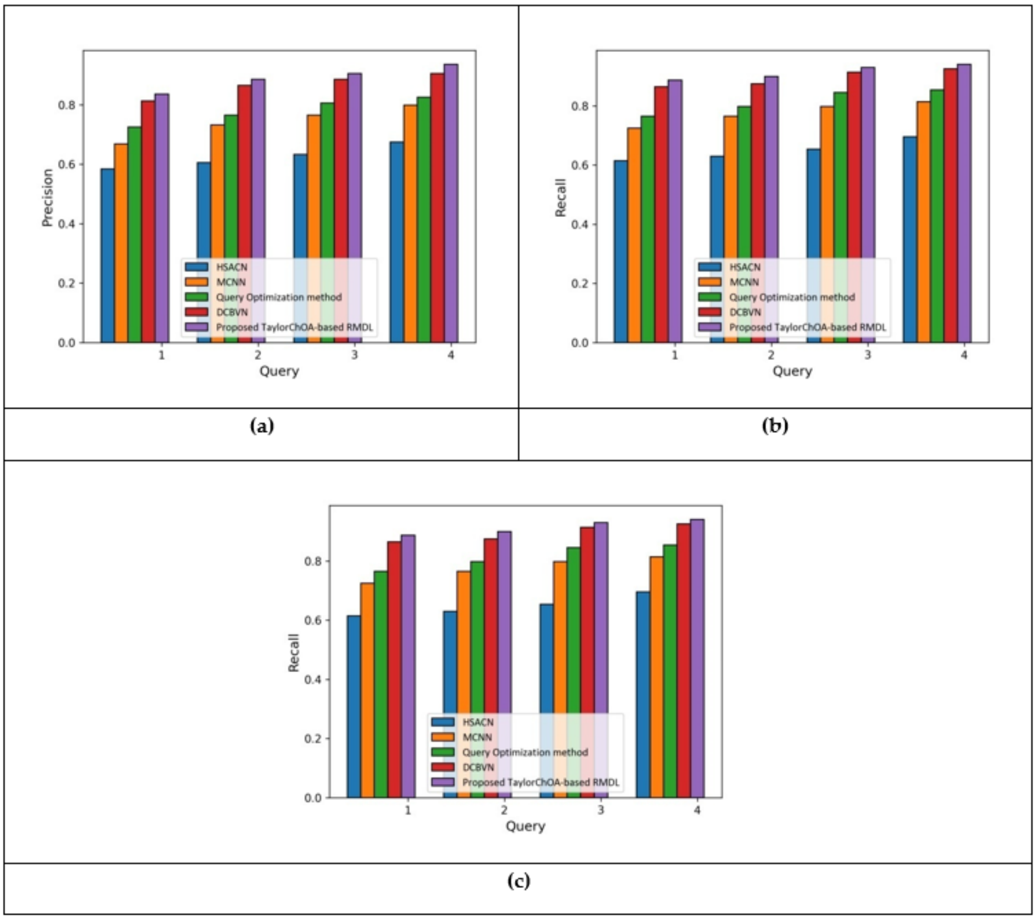
Figure 5. Comparative analysis with cluster size = 3 using K-Khool dataset: (**a**) precision, (**b**) recall, and (**c**) F1-score.

5.1.4. Comparative Analysis Based on Cluster Size = 4 in Terms of Precision, Recall, and F1-Score Using E-Khool Dataset

The analysis with cluster size = 4 using the evaluation metrics and varying the number of queries is portrayed in Figure 6. The analysis using precision is shown in Figure 6a. When considering the number of query as 1, the developed TaylorChOA-based RMDL computed a precision value of 0.836, whereas thepPrecision value achieved by the existing methods, such as HSACN, MCNN, Query Optimization, and DCBVN, is 0.584, 0.668, 0.725, and 0.812, respectively.

Figure 6b presents the assessment using recall. The recall values obtained by the HSACN, MCNN, Query Optimization, DCBVN, and developed TaylorChOA-based RMDL are 0.629, 0.765, 0.798, 0.874, and 0.899, respectively, for the number of query 2. The analysis in terms of recall measure is presented in Figure 6c. When the number of query is 3, the F1-score value of HSACN is 0.643, MCNN is 0.781, Query Optimization is 0.824, DCBVN is 0.899, and developed TaylorChOA-based RMDL is 0.917.

Table 1 presents a comparison of the results developed by the TaylorChOA-based RMDL technique with the existing techniques by considering the evaluation measures for the number of query 4. With cluster size = 3, the maximum precision of 0.925, maximum recall of 0.944, and maximum F1-score of 0.934 are computed by the developed TaylorChOA-based RMDL method.



(a)

(b)

(c)

Figure 6. Comparative analysis with cluster size = 4 using E-Khool dataset: (**a**) precision, (**b**) recall, and (**c**) F1-score.

Table 1. Comparison of proposed TaylorChOA-based RMDL with existing methods using E-Khool dataset, in terms of precision, recall, and F1-score.

Method	Metrics	HSACN	MCNN	Qu Opt.	DCBVN	Proposed Method
Cluster Size = 3	Precision	0.684	0.784	0.814	0.896	**0.925**
	Recall	0.685	0.805	0.854	0.925	**0.944**
	F1-score	0.684	0.794	0.833	0.910	**0.934**
Cluster Size = 4	Precision	0.674	0.798	0.825	0.905	**0.936**
	Recall	0.695	0.814	0.854	0.925	**0.941**
	F1-score	0.685	0.806	0.839	0.915	**0.938**

Using cluster size = 4, the maximum precision of 0.936 is computed by the developed TaylorChOA-based RMDL, whereas the Precision value computed by the existing methods, such as HSACN, MCNN, Query Optimization, and DCBVN is 0.674, 0.798, 0.825, and 0.905, respectively. Likewise, the higher recall of 0.941 is computed by the developed TaylorChOA-based RMDL, whereas the precision value computed by the existing methods, such as HSACN, MCNN, Query Optimization, and DCBVN is 0.695, 0.814, 0.854, and 0.925, respectively. Moreover, the F1-score value obtained by the HSACN is 0.685, MCNN is 0.806, Query Optimization is 0.839, DCBVN is 0.915, and TaylorChOA-based RMDL is 0.938. Thus, the developed TaylorChOA-based RMDL outperformed various existing methods and achieved better performance with the maximum precision of 0.936, maximum recall of 0.944, and maximum F1-score of 0.938.

5.2. Results Based on Coursera Course Dataset with Respect to the Number of Iterations (10 to 50)

5.2.1. Performance Analysis Based on Cluster Size = 3

Figure 7 presents the performance analysis of the developed technique with iterations by varying the queries with cluster size = 3. Figure 7a presents the assessment based on precision.

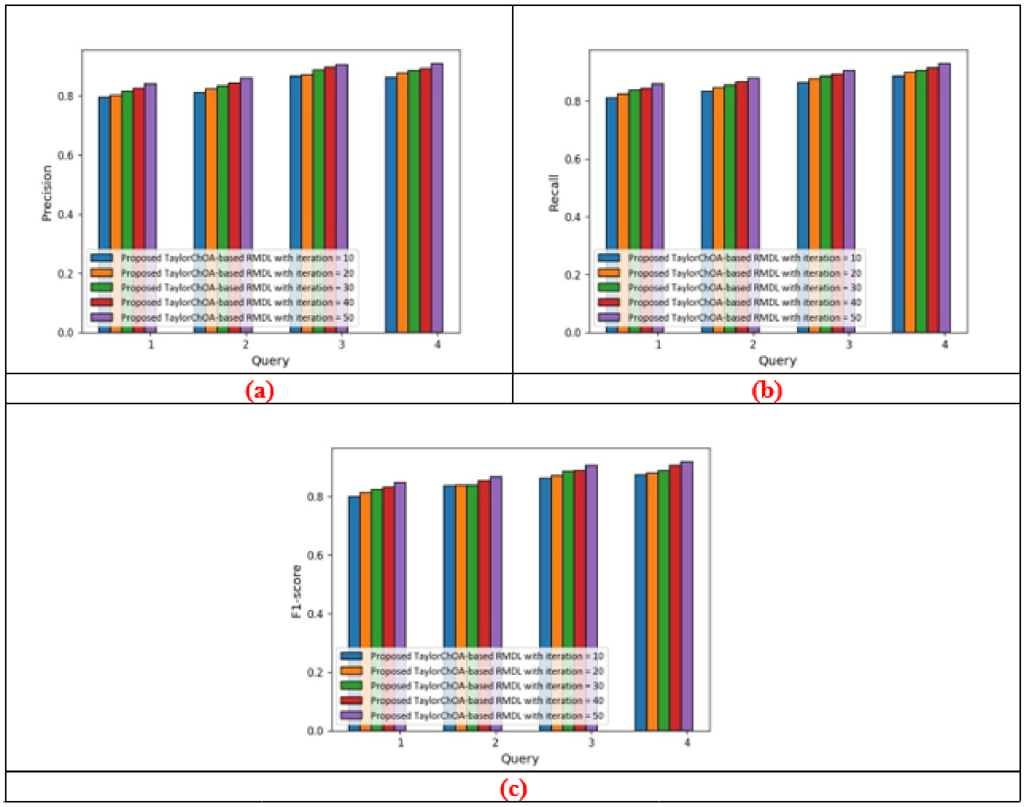


Figure 7. Performance analysis with cluster size = 3 using Coursera Course Dataset: (**a**) precision, (**b**) recall, and (**c**) F1-score.

For the number of query 1, the precision value measured by the developed TaylorChOA-based RMDL with iteration 10 is 0.795, iteration 20 is 0.825, iteration 30 is 0.836, iteration 40 is 0.847, and iteration 50 is 0.854. Figure 7b portrays the analysis using recall. By considering the number of query 2, the value of recall computed by the developed TaylorChOA-based RMDL with iteration 10 is 0.825, iteration 20 is 0.847, iteration 30 is 0.874, iteration 40 is 0.885, and iteration 50 is 0.895. The analysis using F1-score is depicted in Figure 7c. When the number of a query is 3, the value of F1-score computed by the developed TaylorChOA-based RMDL with iteration 10 is 0.863, iteration 20 is 0.871, iteration 30 is 0.886, iteration 40 is 0.890, and iteration 50 is 0.907.

5.2.2. Performance Analysis Based on Cluster Size = 4

Figure 8 presents the performance analysis of the developed technique with iterations by varying the queries with cluster size = 6.

Figure 8a presents the assessment based on precision. For the number of query 1, the precision value measured by the developed TaylorChOA-based RMDL with iteration 10 is 0.804, iteration 20 is 0.815, iteration 30 is 0.825, iteration 40 is 0.831, and iteration 50 is 0.848. Figure 8b portrays the analysis using recall. By considering the number of query 2, the value of recall computed by the developed TaylorChOA-based RMDL with iteration 10 is 0.843, iteration 20 is 0.854, iteration 30 is 0.865, iteration 40 is 0.871, and iteration 50 is 0.888. The analysis using F1-score is depicted in Figure 8c). When the number of a query is 3, the value of F1-score computed by the developed TaylorChOA-based RMDL with iteration 10 is 0.854, iteration 20 is 0.865, iteration 30 is 0.874, iteration 40 is 0.888, and iteration 50 is 0.898.

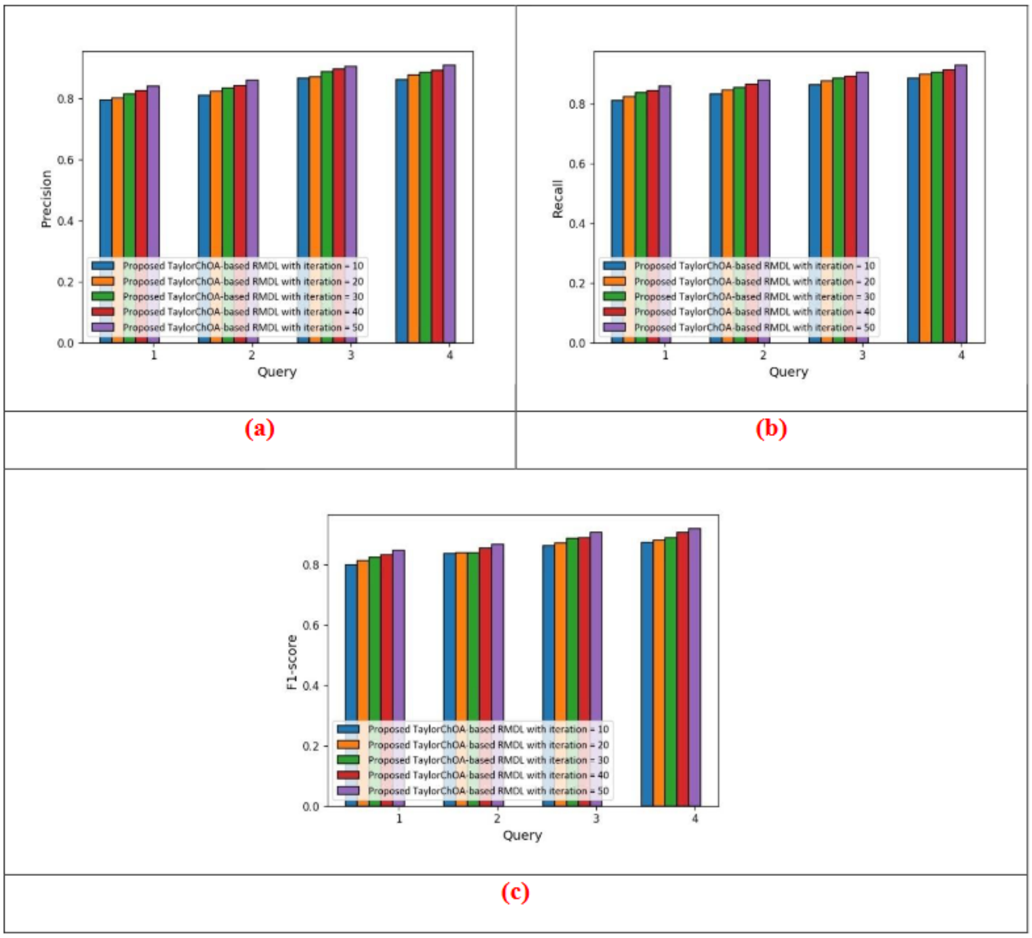


Figure 8. Performance analysis with cluster size =4 using Coursera Course Dataset: (**a**) precision, (**b**) recall, and (**c**) F1-score.

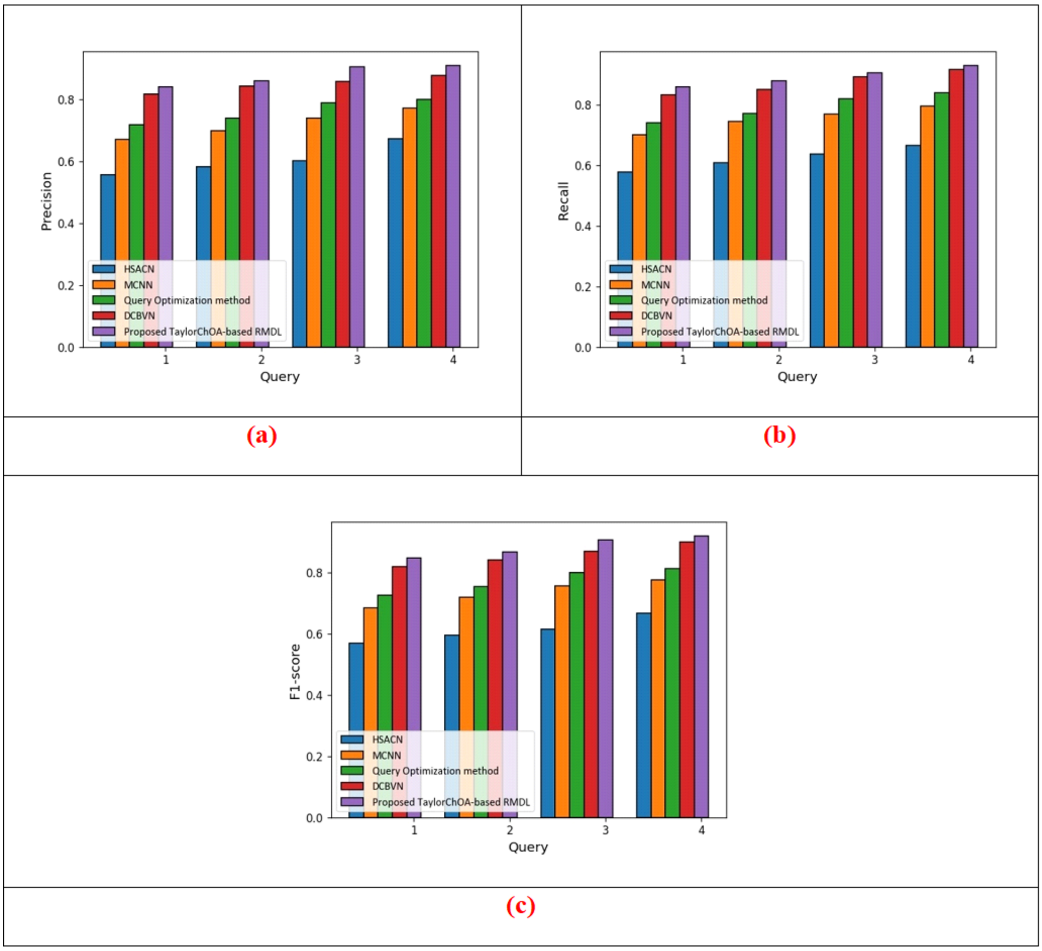
Comparison of existing methods and the proposed TaylorChOA-based RMDL using Coursera Course Dataset, in terms of precision, recall, and F1-Score

The comparative assessment of the developed technique is performed by varying the queries with the cluster size = 3 and cluster size = 4 in terms of the evaluation metrics.

5.2.3. Analysis Based on Cluster Size = 3 in Terms of Precision, Recall, and F1-Score

Figure 9 portrays the assessment with cluster size = 3 by varying the number of queries using the performance measures, such as precision, recall, and F1-score.

Figure 9a presents the analysis in terms of precision. When number of query is 1, the precision value measured by the developed TaylorChOA-based RMDL is 0.839, whereas the precision value measured by the existing methods, such as HSACN, MCNN, Query Optimization, and DCBVN is 0.556, 0.669, 0.716, and 0.816, respectively. The analysis based on recall measure is portrayed in Figure 9b. By considering the number of query as 2, the developed TaylorChOA-based RMDL measured a recall value of 0.878, whereas the value of recall computed by the existing methods, such as HSACN, MCNN, Query Optimization, and DCBVN is 0.606, 0.743, 0.769, and 0.849, respectively. The assessment using F1-score is shown in Figure 9c. The F1-score value attained by the HSACN, MCNN, Query Optimization, DCBVN, and developed TaylorChOA-based RMDL is 0.615, 0.756, 0.800, 0.870, and 0.907, respectively, when considering the number of query as 3.



(a)

(b)

(c)

Figure 9. Comparative analysis with cluster size =3 using Coursera Course Dataset: (**a**) precision, (**b**) recall, and (**c**) F1-score.

5.2.4. Analysis Based on Cluster Size = 4 in Terms of Precision, Recall, and F1-Score

The analysis with cluster size = 4 using the evaluation metrics, by varying the number of queries is portrayed in Figure 10.

The analysis using precision is shown in Figure 10a. When considering the number of query as 1, the developed TaylorChOA-based RMDL computed a precision value of 0.836, whereas the precision value achieved by the existing methods, such as HSACN, MCNN, Query Optimization, and DCBVN is 0.584, 0.668, 0.725, and 0.812, respectively. Figure 10b presents the assessment using recall. The recall values obtained by the HSACN, MCNN, Query Optimization, DCBVN, and developed TaylorChOA-based RMDL are 0.629, 0.765, 0.798, 0.874, and 0.899, respectively, for the number of query 2. The analysis in terms of F-1 score is presented in Figure 10c. When the number of query is 3, the F1-score value of HSACN is 0.643, MCNN is 0.781, Query Optimization is 0.824, DCBVN is0.899, and developed TaylorChOA-based RMDL is 0.917.

Table 2 explains the comparative discussion of the developed Taylor ChOA-based RMDL technique in comparison with the existing techniques using the Coursera Course dataset for the number of query 4. With cluster size = 3, the maximum precision of 0.908, maximum recall of 0.928, and maximum F1-score of 0.919 are computed by the developed Taylor ChOA-based RMDL method. Using cluster size = 4, the maximum precision of 0.919 is computed by the developed Taylor ChOA-based RMDL, whereas the precision value computed by the existing methods, such as HSACN, MCNN, Query Optimization, and DCBVN is 0.667, 0.776, 0.813, and 0.899, respectively. Likewise, the higher recall of 0.926 is computed by the developed Taylor ChOA-based RMDL, and the F1-score value is 0.925.

From this table, it is clear that, the developed Taylor ChOA-based RMDL outperformed various existing methods.

Table 3 shows the computational time of proposed and existing methods for query = 1. The proposed system has the minimum computational time of 127.25 s and 133.84 s for E-Khool dataset, and Coursera Course dataset, respectively.

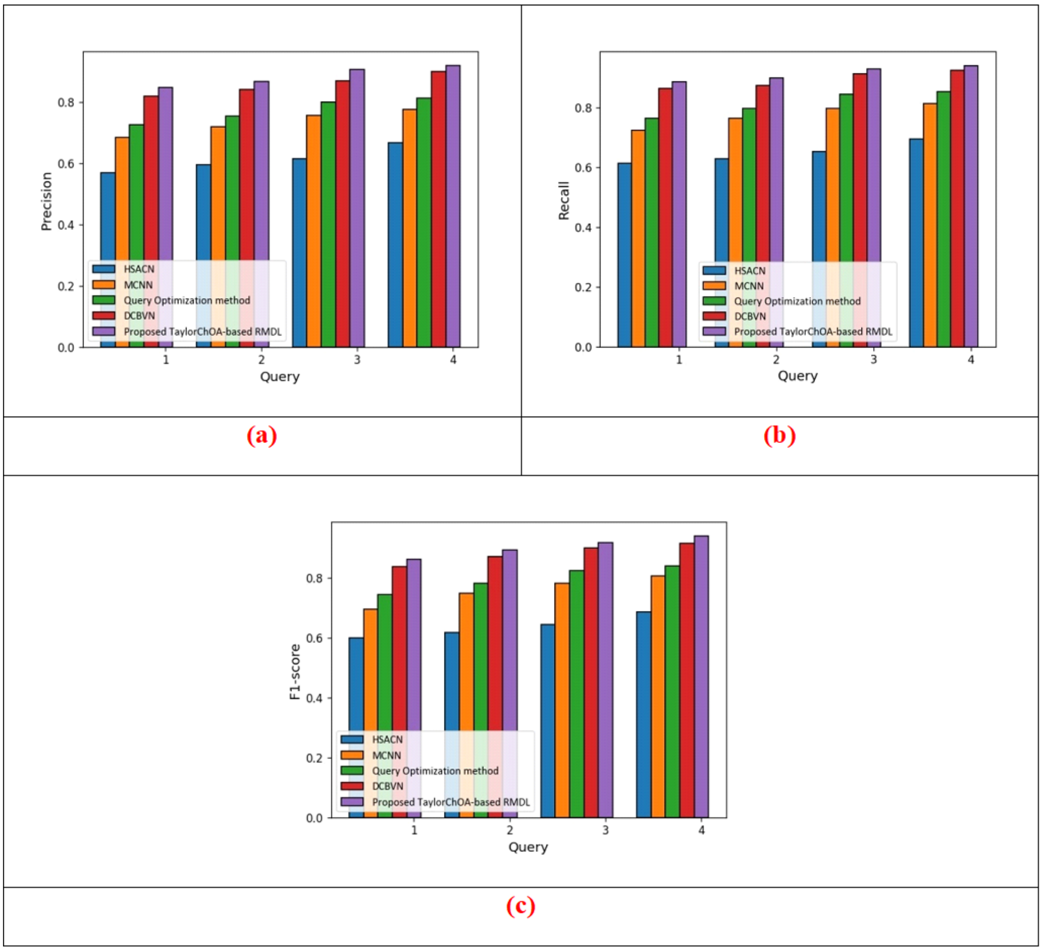


Figure 10. Comparative analysis with cluster size = 4 using Coursera Course Dataset: (**a**) precision, (**b**) recall, and (**c**) F1-score.

Table 2. Comparison of proposed TaylorChOA-based RMDL with existing methods using Coursera Course dataset, in terms of precision, recall, and F1-score.

Method	Metrics	HSACN	MCNN	Qu Opt.	DCBVN	Proposed Method
Cluster Size = 3	Precision	0.672	0.772	0.798	0.877	**0.908**
	Recall	0.665	0.795	0.837	0.914	**0.928**
	F1-score	0.667	0.776	0.813	0.899	**0.919**
Cluster Size = 4	Precision	0.667	0.776	0.813	0.899	**0.919**
	Recall	0.676	0.798	0.839	0.907	**0.926**
	F1-score	0.674	0.788	0.825	0.899	**0.925**

Table 3. Comparison of computational time, in terms of seconds.

Dataset	Time	HSACN	MCNN	Qu Opt.	DCBVN	Proposed Method
E-Khool	Seconds	182.41	180.41	162.25	145.36	**127.25**
Coursera Course	Seconds	192.45	187.52	170.54	153.25	**133.84**

6. Conclusions and Future Work

This research aims to resolve the problem of information overload in the online education field. Choosing a personalized course on an online education website may be extremely difficult and tedious. Hence, this research proposes a robust sentiment classification model to recommend the courses using the proposed TaylorChOA-based RMDL method. Here, a course review is performed by considering the review data for finding the best course. With the best course, various features, such as SentiWordNet-based statistical features, classification-specific features, and TF-IDF features, are effectively extracted from the review data. After the extraction of significant features, the RMDL model is used to classify the sentiments, and the training practice of RMDL is done using the developed optimization algorithm, known as Taylor ChOA. Thus, the course recommendation is done by offering positively recommended courses to the user. TaylorChOA is newly designed by the combination of the Taylor concept and the ChOA algorithm. Moreover, the developed technique attained better performance using precision, recall, and F1-score with the higher values of, 0.936, 0.944, and 0.938, respectively. However, the performance of the devised approach is not evaluated using more evaluation metrics. In the future, the developed work can be further extended by developing deep learning classifiers and evaluating the performance using more evaluation metrics.

Author Contributions: S.K.B. designed and wrote the paper; H.L. supervised the work; S.K.B. performed the experiments with advice from B.X.; and D.K.S. organized and proofread the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used in the experiments are publicly available. Details have been given in Section 4.1.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ChOA	Chimp Optimization Algorithm
DCBVN	Demand-aware Collaborative Bayesian Variational Network
DÉCOR	Deep learning-enabled Course Recommender System
DNN	Deep Neural Networks
GRU	Gated Recurrent Unit
HANCI	Hierarchical Attention Network Oriented towards Crowd Intelligence
HSACN	Hierarchical Self-Attentive Convolution Network
LSTM	Long Short-Term Memory
MCNN	Multi-model Convolutional Neural Network
NLP	Natural Language Processing
RMDL	Random Multi-model Deep Learning
RNN	Recurrent Neural Network

References

1. Wen-Shung Tai, D.; Wu, H.-J.; Li, P.-H. Effective e-learning recommendation system based on self-organizing maps and association mining. *Electron. Libr.* **2008**, *26*, 329–344. [CrossRef]
2. Persky, A.M.; Joyner, P.U.; Cox, W.C. Development of a course review process. *Am. J. Pharm. Educ.* **2012**, *76*, 130. [CrossRef]
3. Guanchen, W.; Kim, M.; Jung, H. Personal customized recommendation system reflecting purchase criteria and product reviews sentiment analysis. *Int. J. Electr. Comput. Eng.* **2021**, *11*, 2399–2406. [CrossRef]
4. Gunawan, A.; Cheong, M.L.F.; Poh, J. An Essential Applied Statistical Analysis Course using RStudio with Project-Based Learning for Data Science. In Proceedings of the 2018 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE), Wollongong, Australia, 4–7 December 2018; pp. 581–588.

5. Assami, S.; Daoudi, N.; Ajhoun, R. A Semantic Recommendation System for Learning Personalization in Massive Open Online Courses. *Int. J. Recent Contrib. Eng. Sci. IT* **2020**, *8*, 71–80. [CrossRef]
6. Hua, Z.; Wang, Y.; Xu, X.; Zhang, B.; Liang, L. Predicting corporate financial distress based on integration of support vector machine and logistic regression. *Expert Syst. Appl.* **2007**, *33*, 434–440. [CrossRef]
7. Aher, S.B.; Lobo, L. Best combination of machine learning algorithms for course recommendation system in e-learning. *Int. J. Comput. Appl.* **2012**, *41*. [CrossRef]
8. Tarus, J.K.; Niu, Z.; Mustafa, G. Knowledge-based recommendation: A review of ontology-based recommender systems for e-learning. *Artif. Intell. Rev.* **2018**, *50*, 21–48. [CrossRef]
9. Zhang, H.; Huang, T.; Lv, Z.; Liu, S.; Zhou, Z. MCRS: A course recommendation system for MOOCs. *Multimed. Tools Appl.* **2018**, *77*, 7051–7069. [CrossRef]
10. Li, Q.; Kim, J. A Deep Learning-Based Course Recommender System for Sustainable Development in Education. *Appl. Sci.* **2021**, *11*, 8993. [CrossRef]
11. Almahairi, A.; Kastner, K.; Cho, K.; Courville, A. Learning distributed representations from reviews for collaborative filtering. In Proceedings of the 9th ACM Conference on Recommender Systems, Vienna, Austria, 16–20 September 2015; pp. 147–154.
12. Yang, C.; Zhou, W.; Wang, Z.; Jiang, B.; Li, D.; Shen, H. Accurate and Explainable Recommendation via Hierarchical Attention Network Oriented Towards Crowd Intelligence. *Knowl.-Based Syst.* **2021**, *213*, 106687. [CrossRef]
13. Zheng, L.; Noroozi, V.; Yu, P.S. Joint deep modeling of users and items using reviews for recommendation. In Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, Cambridge, UK, 6–10 February 2017; pp. 425–434.
14. Zeng, H.; Ai, Q. A Hierarchical Self-attentive Convolution Network for Review Modeling in Recommendation Systems. *arXiv* **2020**, arXiv:2011.13436.
15. Dong, X.; Ni, J.; Cheng, W.; Chen, Z.; Zong, B.; Song, D.; Liu, Y.; Chen, H.; De Melo, G. Asymmetrical hierarchical networks with attentive interactions for interpretable review-based recommendation. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 7667–7674.
16. Wang, H.; Wu, F.; Liu, Z.; Xie, X. Fine-grained interest matching for neural news recommendation. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Seattle, WA, USA, 5–10 July 2020; pp. 836–845.
17. Bansal, T.; Belanger, D.; McCallum, A. Ask the gru: Multi-task learning for deep text recommendations. In Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, 15–19 September 2016; pp. 107–114.
18. Tay, Y.; Luu, A.T.; Hui, S.C. Multi-pointer co-attention networks for recommendation. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018; pp. 2309–2318.
19. Bai, Y.; Li, Y.; Wang, L. A Joint Summarization and Pre-Trained Model for Review-Based Recommendation. *Information* **2021**, *12*, 223. [CrossRef]
20. Da'u, A.; Salim, N.; Rabiu, I.; Osman, A. Recommendation system exploiting aspect-based opinion mining with deep learning method. *Inf. Sci.* **2020**, *512*, 1279–1292.
21. Wang, C.; Zhu, H.; Zhu, C.; Zhang, X.; Chen, E.; Xiong, H. Personalized Employee Training Course Recommendation with Career Development Awareness. In Proceedings of the Web Conference 2020, Taipei, Taiwan, 20–24 April 2020; pp. 1648–1659.
22. Rafiq, M.S.; Jianshe, X.; Arif, M.; Barra, P. Intelligent query optimization and course recommendation during online lectures in E-learning system. *J. Ambient. Intell. Humaniz. Comput.* **2021**, *12*, 10375–10394. [CrossRef]
23. Sulaiman, M.S.; Tamizi, A.A.; Shamsudin, M.R.; Azmi, A. Course recommendation system using fuzzy logic approach. *Indones. J. Electr. Eng. Comput. Sci.* **2020**, *17*, 365–371. [CrossRef]
24. Xie, J.; Girshick, R.; Farhadi, A. Unsupervised deep embedding for clustering analysis. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; pp. 478–487.
25. 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, Lakeland, FL, USA, 9–1 April 2018; pp. 19–28.
26. Mangai, S.A.; Sankar, B.R.; Alagarsamy, K. Taylor series prediction of time series data with error propagated by artificial neural network. *Int. J. Comput. Appl.* **2014**, *89*, 41–47.
27. Khishe, M.; Mosavi, M.R. Chimp optimization algorithm. *Expert Syst. Appl.* **2020**, *149*, 113338. [CrossRef]
28. Ohana, B.; Tierney, B. Sentiment classification of reviews using SentiWordNet. In Proceedings of the IT&T, Dublin, Ireland, 22–23 October 2009.
29. Christian, H.; Agus, M.P.; Suhartono, D. Single document automatic text summarization using term frequency-inverse document frequency (TF-IDF). *ComTech Comput. Math. Eng. Appl.* **2016**, *7*, 285–294. [CrossRef]