A Hybridized Deep Learning Strategy for Course Recommendation

Gerard Deepak, National Institute of Technology, Tiruchirappalli, India*

Ishdutt Trivedi, Manipal Institute of Technology Bengaluru, Manipal Academy of Higher Education, Manipal, India

ABSTRACT

Recommender systems have been actively used in many areas like e-commerce, movie and video suggestions, and have proven to be highly useful for its users. But the use of recommender systems in online learning platforms is often underrated and less likely used. But many of the times it lacks personalisation especially in collaborative approach while content-based doesn't work well for new users. Therefore, the authors propose a hybrid course recommender system for this problem which takes content as well as collaborative approaches and tackles their individual limitations. The authors see recommendation as a sequential problem and thus have used RNNs for solving the problem. Each recommendation can be seen as the new course in the sequence. The results suggest it outperforms other recommender systems when using LSTMs instead of RNNs. The authors develop and test the recommendation system using the Kaggle dataset of users for grouping similar users as historical data and search history of different users' data.

KEYWORDS

Course Recommendation, Hybridized Strategy, LSTM Spectral Clustering

1. INTRODUCTION

In this fast-paced digital world, everyone is drifting towards electronic resources for getting their stuff done, be it books, movies, or an entire learning system. The term E-Learning is a formal teaching methodology which uses electronic resources as their parameter. It just requires you an internet connection which is not difficult to find in this digitalized world. E-Learning is nowadays very popular among people as it allows them to learn the topics from the best-in-class faculties in the world without any discrimination. Unlike traditional learning which requires you to sit in a class for a fixed amount of time, you can study through E-Learning from any place you want and at any pace. It provides a lot of conveniences, that's why it is widely adopted everywhere.

With an increasing population dependent on electronic resources for reading, it would be highly convenient if the website or app itself recommend you the new topics/courses based on specific attributes so that you don't have to search for it every time. Many a time since the E-Learning platform

DOI: 10.4018/IJAET.321752

*Corresponding Author

is not well organized, it isn't straightforward for a student to find the next appropriate course. There are lots of advice you can find on the internet, and most of the time it confuses the student. It was a significant disadvantage of E-Learning which is why people still relied on traditional classroom learning methods as their teacher used to tell what you should study after this course. If studied in a mismatch course, It can lead to a lack of motivation for a student to study the entire subject sometimes.

This problem is solved by a course recommender system which takes various parameters into account and then suggests the course. A course recommender system is a subset software based on information filtering concept. It customizes the needs of a student and shows the most relevant courses for an individual and thus creating a personal learning environment. It uses efficient information retrieval techniques for this which can be seen in many other fields too, like Netflix (movie recommendation), YouTube (video recommendation), eCommerce sites.

Based on the various parameters taken into account for the recommendation, it can be divided into three categories: content-based, collaborative, and knowledge-based. Collaborative recommender system takes preferences that people that have a similar liking in the past have a similar liking in future too. The concept of content-based systems is based on the presumption that people who like an item with a particular attribute will also like the same attributes in the future while knowledge-based takes the data of a person to recommend it the suggestions. It is more accurate but takes a lot of data from the consumer, and hence in this privacy concerned world, it is not preferred much. There is one more category of recommendation system, which is formed by combining two or more above types in order to maximize the accuracy of suggestion and reducing the disadvantages. This is called the Hybrid recommender system. This research will focus on Hybrid recommender system.

The following are the novel Propositions in the Proposed Work:

1. The usage of the Web Usage Data of the user requesting the courses for recommendation and also the collective requirements and interests of similar users are taken into consideration.
2. Using a Spectral Clustering technique for grouping the profiles of similar users such that a collective intelligence of individual user profiles can be harvested.
3. The usage of LSTM by appending the features from Semantic Networks formulated based on the User Query and Current User Clicks and Enriching it based on the real-world knowledge form Wikidata is one of the novel contributions.
4. Also, collectively imbibing both clustering and a classification into a single framework and transforming the approach based on knowledge harbored from the external knowledge stores makes it quite novel.

The remainder of the paper is formatted as follows. Section 2 illustrates the Related Work. In Section 3, the problem definition is explained. Section 4 depicts the Proposed Methodology. Section 5 discusses implementation and performance evaluation. Section 6 brings the paper to a conclusion.

2. RELATED WORK

2.1 Recommendation System Using Deep Learning

Hu et al. (2019) have proposed an approach using attention incorporated graph convolutional networks to predict the performance of the student. The model can capture the relational structure that underpins the data in students' course records. The accuracy of the model's grade prediction and its capacity to recognize at-risk children were also tested.

Sultana et al. (2019) have suggested a Knowledge Discovery technique using twitter data (educational tweets) that was carried out in this research work utilizing deep neural networks. Positive, negative, and neutral data were used to classify the information. Deep Learning methods were applied to training data. The modeling was achieved and evaluated using newly formed test data, with a only few variables being assessed. Bhumichitr et al. (2017) have suggested a recommendation system for university elective subjects that suggests courses based on similarities across students' course templates. Two common algorithms were used in this study: collaboration based recommendation utilising the Pearson Correlation Coefficient and Alternating Least Squares (ALS).

Nassar et al. (2020) used deep learning, they developed a multi-criterion driven approach infusing the concept of collaborative filtering. It has a pair of functionalities, the former for prediction of the criterion and the later for prediction of the total rating. More complicated neural network topologies, such as the Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN), can be used to improve it even more. For a collaborative filtering recommender system. Bobadilla et al. (2020) suggested an equitable model that classifies using deep neural network. The binary voting methodology has been incorporated for learning considering relevant and non-relevant items. This data reduction leads to a new degree of abstraction, and classification-based architecture emerges as a result. Buhagiar et al. (2018) have proposed a model based on the analysis of the comments depicted in Reddit discussion forums. The probabilistic topic model Latent Dirichlet Allocation (LDA) was used to identify the chat subjects. Using these subjects as features for a neural network, several different neural network architectures were trained on the data to function as models for identifying which threads a specific user would be interested in adding to based on their previously indicated interests.

Using a parallel method, Hong et al. (2020) suggested a Cross Domain Deep Neural Network (CD-DNN) in order to facilitate recommendations across inter-related domains. The proposition can solve the problem involving the prediction of ratings modelling persons and items on the basis of item metadata and reviews. Instead of only collaborative or content-based filtering, Pornwattanavichai et al. (2020) proposed a novel technique of promoting Tweets based on hybridized rrecommendation with Latent Dirichlet Allocation and matrix factorization. Zhu et al. (2020) suggested a neural circuit mechanism for performing a task associated with context dependence, which needs connecting sensory stimulusus for behavioral responses in order to generalize to various symmetric situations. This approach employs gated neural units to control the circuit's physiological connection pattern.

By bridging the gap associated with neural networks and factorization models, Jiang et al. (2020) presented an efficient and equitable recommendation approach to tackle the additive user problem. The efficient portion is mechanism that deals with constraint correlation using factorization, while the equitable portion is a neural network which encodes, compresses, and fuses several multiple. The recommender integrates the two elements so that the factorization model and the neural network can operate together. For recommender systems, Chen et al. (2019) introduced the Joint Neural Collaborative Filtering (J-NCF) approach. The J-NCF model incorporates a neural network with a rating matrix that combines deep learning of features and interaction modelling using deep learning. Incorporating paradigms of deep learning architecture based on a user-item, a matrix of ratings composing of deep feature learning using representation of features consisting of people and objects.

2.2 Recommendation System Using Clustering

Asadi et al. (2019) presented a recommender model that considers student attributes while making course recommendations. Clustering was utilized in the model to find students with similar interests and abilities. Following the discovery of comparable students, fuzzy association rules mining was used to investigate the connections between student course selections. Clustering and fuzzy association algorithms are used to provide a suitable suggestion and a projected score. Moubayed et al. (2020) proposed using the k-means approach for clustering of students on the basis of twelve engagement indicators separated based on effort and interaction. Quantitative analysis was used to identify pupils who were not engaged and may want assistance. Two-level, three-level, and five-level clustering models were investigated. MATLAB was used to convert the event log into a new dataset that represented the measures under consideration.

Gulzar et al. (2018) presented a recommender system that suggests and assists a user in selecting courses that meet their needs. To obtain valuable information and generate correct suggestions, they utilized a hybrid technique that included ontology. Users' performance and satisfaction may improve as a result of this level. Al-(Badarenah et al., 2016) developed a novel recommendation approach that used association rules for suggesting university course electives for targeting a student on the basis of other students with similar interest. As a result of the trials, the association rule has shown to be a popular technique for delivering a suggestion to a target student. They discovered patterns of effect of various factors on the system's performance because of their investigation. Ognjanovic et al. (2016) proposed a method for extracting student preferences from institutional student information systems sources. The Analytical Hierarchy Process (AHP) was used to analyse the retrieved alternatives and forecast student course selection. Using the dataset gathered in an undergraduate degree programme at a research focused institution in Canada (N = 1061), the AHP-based method was verified.

2.3 Traditional Recommendation System

Traditional focus on user commonality may be exaggerated, according to (O'Donovan et al., 2005) . Additional criteria, they claimed, play a substantial influence in influencing suggestions. They focused on user trustworthiness, presenting two computational trust models, and demonstrating how they might be easily integrated into common collaborative filtering frameworks in a number of ways. According to Siewe et al. (2019), a unique and successful model for recommendation approach that offers tailored learning items on the basis of student's style of learning has been presented. The method is built on Felder and Silverman's model for learning style, that describes both learning object profiles and student profile. Using motivation from multi-label classification and mixture models, Gruver et al. (2019) developed a probabilistic way to modelling course enrollment decisions. They built a model for learning on the basis of joint distribution that uses a latent Gaussian variable model for learning 10 years of anonymized student records from a big institution. The model supports a wide range of inference queries which is resistant too sparsity in data.

For finding the most important factors that might affect the optional course suggestion for university students, Esteban et al. (2018) presented a several criteria strategy on the basis of integration Collaborative Filtering (CF) and Content Based Filtering (CBF). It presented a genetic algorithm that automatically finds the relevance of the different criteria and assigns weights to each of them in order to understand which aspects are the most essential. Polyzou et al. (2019) put forth a novel strategy for extracting sequenced patterns from prior course enrolment data for providing a list of tailored course choices for the subsequent semester. The suggested technique used a methodology employing random walks on a graph mapping courses, and customization was accomplished using a student-adapted beginning distribution based on current student enrollments. Bhattacharya et al. (2018) have suggested an approach for recommendation system in assisting new workers by forecasting the qualifications and skills they should get to succeed in their professions based on their personal historical skills, education and certification. For this purpose, they used Trees with Compact Prediction, a relatively new method.

The literature of the existing works have been analysed and it is very clear that either clustering techniques have been used or classification techniques have been used separately. There is a scope when systematically clustering and classification is segregated in the framework. Moreover, spectral clustering has not been used for a course recommender system so far. So as a result when spectral clustering is incorporated, it is clear that the user's similarity among a collective group of users is deduced and only the users with similar recommendation requirements are clustered and that care is taken that spectral clustering clusters are not only based on user's previous visit terms but also the structural aspects of the individual terms. The use of LSTM, when combined with semantic network and collective requirements of user's intelligence is more clear. Furthermore, clustering and classification are not utilised in isolation to provide recommendations. Clustering is used to categorise users based on their search history, whereas classification is primarily used to categorise collective intelligence users using real-world information from Wiki data. Moreover, Wikidata is chosen as a knowledge repository for incorporating real-world knowledge and reducing the cognitive gap between the standard existing real-world knowledge and auxiliary knowledge fed into the framework.

3. PROBLEM DEFINITION

Let S represent all of the students, and C represent all of the courses available. We assume that any student can access any course as this solution will be applicable for online as well as offline learning modes. Our task is to take a student as input and recommend him courses $c_1, c_2 \ldots c_k$ based on parameters like his profile and search history. Since the person is most likely to take the course which he explored about, more bias is to the search history than the profile.

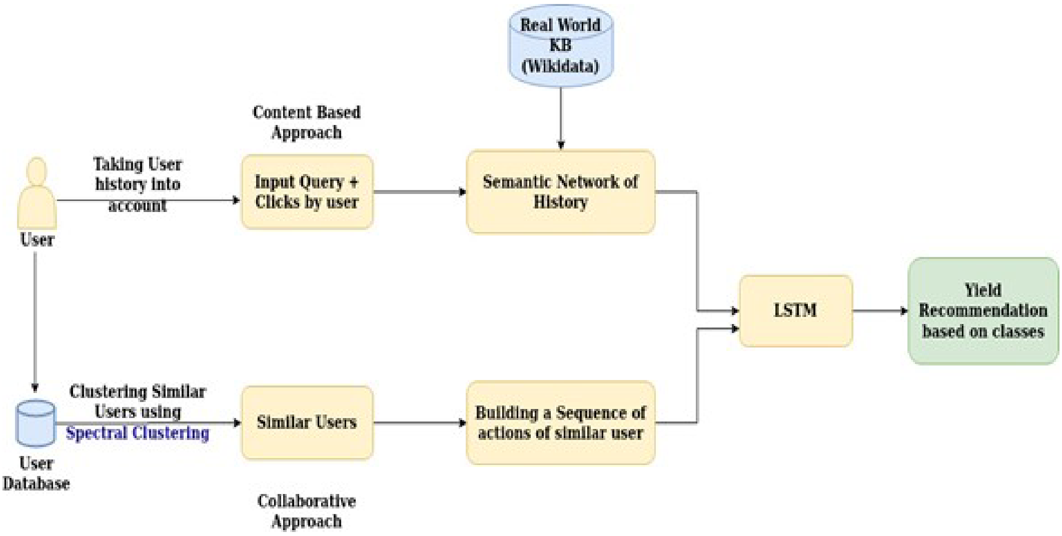
We propose hybrid course recommendation techniques in this combining the collaborative as well as content-based recommender system. This problem can be visualized as a sequence modelling problem when every action is treated as a sequence, and thus Recurrent neural network is employed for solving this problem.

4. PROPOSED METHODOLOGY

4.1 Proposed Architecture

To obtain the recommendation, the suggested model's architecture includes the following primary working components and stages. Fiq.1. shows the pictorial representation of the architecture. The historical content of the user, the web usage data is taken into consideration. The reason for considering web usage data is to imbibe personalization as well as user awareness and user centeredness into the approach. So the input user query along with the current user query is taken into consideration and is subject to pre-processing. Pre-processing involves tokenization, lemmatization, stop word removal and named entity recognition. A semantic network is formulated from the pre-processed data. It is formulated using information content of individual words and semantic similarity which is dynamically computed. To compute semantic similarity, a simple concept similarity is taken into consideration. Wikidata is considered for the real-world knowledge base. The reason for including Wikidata is to enhance ground

Figure 1.
Proposed system architecture


Real World KB (Wikidata)
Content Based Approach
Taking User history into account
User
Input Query + Clicks by user
Semantic Network of History
LSTM
Yield Recommendation based on classes
Clustering Similar Users using Spectral Clustering
User Database
Similar Users
Building a Sequence of actions of similar user
Collaborative Approach

truth and incorporate more entities that are relevant to the user derived entities. In order to access Wikidata, an API is incorporated. Increasing the size of the semantic network ensures that the data density becomes extensively high and a large amount of auxiliary knowledge is provided into the approach.

A similar user clustering takes place, instead of considering the web usage data of single users, web usage data of multiple users are taken into consideration to develop the users' collective consciousness and obtain a consensus on specific topics. Similar user clustering takes place using the spectral clustering approach. Spectral clustering was chosen because it is very robust and can solve a wide range of problems, such as intertwined spirals. It works on huge data sets as well, as long as the similarity matrix is sparse. When similar users are clustered together, a sequence of actions of similar users i.e topic-relevant terms is taken into built which is then fed into LSTM along with the semantic network history. LSTM classifies the content based on the input dataset. The top 10% of classified is furnished to the user under every class at the first instance. Based on the user click, the next 10% are also permuted, until there are no user clicks that are recorded. Top 50 recommendation under each class is sent to the user until the user is satisfied. If the user is satisfied with the top 10% recommendation, then further recommendation is stopped.

4.2 Spectral Clustering

Clustering is a famous unsupervised learning algorithm which is used to group similar data. Spectral clustering is a popular and easy to implement clustering algorithm. It has gained its popularity since it outperforms many other traditional clustering algorithms, including the most popular K-Means. Unlike K-means, which always produces a convex set, spectral clustering can address a wide range of problems, including entangled spirals. It works with huge data sets as well, if similarity matrix is sparse. It is a technique with its base from Graph Theory, where it treats every data point as a node in a graph and thus the problem into a graph partitioning problem. The basic idea is to divide the graph into two parts: edges between separate groups with a low weight (indicating that they are dissimilar and belong to a different cluster) and edges with a high weight (indicating that they are similar and belong to the same cluster). When seen from a graph's perspective, it is just a min-cut problem where the number of cuts is K. This algorithm can be breakdown into 3 steps.

4.2.1 Building the Similarity Matrix

This step includes the construction of a Similarity graph in the form of an adjacency matrix. If the similarity Sij between the selected points X_1 and X_2 is positive and the edges have a weight $Wij = Sij$, two vertices in the similarity graph are connected. There are several algorithms for transforming data points into a graph. The epsilon-neighborhood graph, completely connected graph and the K-nearest neighbour graph, are examples of this.

4.2.2 Reducing the dimensionality of Data

This stage involves reducing the data's dimensionality to K, where K denotes the number of clusters. It uses the concept of the Graph Laplacian Matrix. Laplacian matrix is given by the Equation (1):

$$L = D - W \tag{1}$$

where L denotes Laplacian Matrix, D is Degree Matrix and W is Weight Matrix

Then after computing the Graph Laplacian Matrix, first K eigenvectors are stacked as columns to form a new matrix. This is how the matrix's dimension is reduced.

4.2.3 Clustering the Data

This step includes clustering of reduced data using classic clustering algorithms such as K-means. Suppose the value of K is 2 then consider the values of 2nd Eigenvector, check the values, the positive will be one cluster and the negative will be another.

4.3 Long Short-Term Memory (LSTM)

Recurrent neural networks solves the problem of handling the sequential data by taking input from previous cells and thus taking account of prior input for predicting the next input. It is widely used in text processing, music composition as well as time series prediction. But the major drawback for using Recurrent neural networks is that it is very difficult for learning long term dependencies using gradient descent. Conventional back-propagation for calculating the Stochastic gradient descent involves chain multiplication of all the terms.

When the differentiation is huge, it grows and grows, resulting in an exploding gradient problem, but vice versa results in a vanishing gradient problem. Long short term memory (LSTM) was developed to overcome the problem of vanishing gradients. It takes account of long term dependencies and solves the problem of vanishing gradient by using multiples gates for scaling the output. LSTM can filter out unnecessary information as well as add some extra pieces of information. The core concept of LSTM is cell state, which flows through the cell with only slight linear interaction. Gates skims through some of the information that isn't necessary. A sigmoid neural net layer and a pointwise multiplication operation are also included in LSTM. It has many extra moving components and hence has many hyperparameters that affect its performance significantly. This includes the use of word embeddings, loss optimization methodology like stochastic gradient descent, dropouts for avoiding overfitting, number of layers, recurrent units and mini-batch size.

4.4 Tang Index

It is a more advanced variant of the Xie-Beni (XB) index, in which following functions $\frac{1}{c+1}\sum_{i=1}^{c}\sum_{p=1,p\neq i}^{c} V_i - V_p$, $\frac{1}{c}$ were introduced to evaluate the fuzzy clustering methods, in numerator and denominator of the XB index, respectively. Both of the terms that have been introduced are functions that are punitive. The index's lowest value suggests higher clustering performance. Tang index is defined by Equation 2:

$$V_T\,(\mathrm{U,V;X}) = \frac{\sum_{i=1}^{c}\sum_{k=n+1}^{n+m}\overset{\frown}{\mu}_{ik}^{2}\left\|X_k - V_i\right\|^2 + \frac{1}{c\left(c+1\right)}\sum_{i=1}^{c}\sum_{p=1,p\neq i}^{c}\left\|V_i - V_p\right\|^2}{min\left\|V_i - V_p\right\|^2_{i\neq p} + \frac{1}{c}} \tag{2}$$

$\forall i \neq p, i\, and\, p \in \left\{1,\ldots,c\right\}$. Here X = { $x_1, x_2, x_3, \ldots\ldots, x_n$ } in p-dimensional space, represents a set of n data points. V = [$v_1, v_2, \ldots\ldots, v_c$] denotes the cluster center matrix of the p × n data matrix X and U = $\left[\mu_{ij}\right]_{c\times n}$ represents membership matrix, where x_j associated with v_i whose membership value is linked with μ_{ij}.

4.5 Semantic Network

A semantic network is a structured representation of knowledge that may be used to make inferences and reasoning. Since the first use of semantic networks, a plethora of theories, models, and practical applications for developing and utilising them have arisen from many sectors of academia and industry.

The semantic network is made up of three parts: a) A syntax that specifies the different sorts of nodes and edges that can be taken into account. b) Definition of the meaning or semantics that the nodes, edges, and network as a whole can express. c) The rules of inference.

Nodes, also known as Concepts, and edges, are data structures used in semantic networks. An abstract representation of the ideas, concepts, and units of information that humans create in their minds is referred to as a concept. When a concept has a natural language description, the word or sequence of words that describes the idea becomes the label for that node. If edges in semantic networks are typed, the edge type depicts the nature of the connection between the nodes. Edges, on the other hand, represent some important connections between the concepts.

Individual and communal information acquisition, organisation, management, and utilisation are all aided by semantic networks. Reasoning and inference on the network data are used to extract knowledge from semantic networks. Because semantic networks were created to represent what a piece of information means, the knowledge derived from them does not have to be factually correct or logical. This interpretation of some data may differ from the truth-conditional content or the most likely interpretation. Secondly, depending on the data used to build the network, a semantic network can represent universal, culturally dependent, domain-specific knowledge.

4.6 Dataset Collection

Historical data is collected from Kaggle, which contains information regarding students and the marks obtained by them in general subjects like Maths, Reading, and Writing in their high school. It includes other attributes like gender, ethnicity, parental level of education. All of this will aid in the identification of students with comparable histories, and students with similar backgrounds are more likely to study relevant courses in college. The dataset contains eight columns and includes data from 1000 students. Link to the dataset: https://www.kaggle.com/spscientist/students-performance-in-exams To get the search history of the users, this experiment was conducted on 75 students and their history, which includes the courses they opened as well as the courses they searched for.

4.7 Data Preprocessing

This involves the removal of punctuations, common stop words like I, have, am, at. This step is significant because if fed the data with the unnecessary words too, our neural networks will end up with tons of redundant data which won’t contribute much for predictions but increase the computational complexity. This step includes tokenization, stemming and lemmatization. Tokenization refers to the conversion of a sentence into small pieces called tokens at the same time removing the punctuation. It can be done easily by just parsing through the sentences. Stemming and lemmatization include identification and removal of words with different forms of a common base. Stemming removes the ends of the word in a hope to the base word. Lemmatization involves the use of vocabulary and morphological analysis of the word to remove the inflectional ending. Porter’s algorithm is the most often used approach for stemming English words.

4.8 Building Semantic Network

It includes building a semantic network from the user’s search history, which consists of the courses opened as well as the input query typed by the user. The main aim for this is to produce a network which is fed to Long Short Term Memory with some bias so that it can be interpreted and the following output, that is course can be predicted for the target user. The network provides a structured representation of text which can be easily used as an input for further processing.

We employed a definitional semantic network, which takes plain text as input and constructs a network by converting each sentence into a network fragment based on semantic analysis. The network is developed in stages, with each sentence being parsed, translated into a short sentence, and then linked to the central network created from all of the preceding sentences. ASKNET was used for this purpose.

4.9 Clustering

This component aims to group similar users(users with similar grades, interests, college, future goals). The core idea is that similar users may have similar course preference. This is referred to as a content-based recommendation system. We intend to use this cluster to build a sequence of action of similar users which is then fed to Long short term memory. This introduces variation in the recommendations and works even for a new user with no past search history. This step runs parallel with building the semantic network to generate a sequence of actions. Since this approach leads to more varied results, LSTM is more biased towards the search results network to minimize the variance. We have used a modified spectral clustering approach which uses the entropy of the system as well as tang index for validating the accuracy of it. Following is the detailed algorithm depicted as Algorithm 1.

4.10 LSTM

The input from the previous stages is fed into LSTM to get the sequence of output. More bias is added to users search history compared to courses based on similar users as the user is more likely to take the course which he explored in the past, Recurrent neural networks can be used to get the next sequence as a recommendation because they are both sequences. For optimizing the results, various hyperparameters have been checked for LSTM to produce an optimized solution for our problem. These include setting appropriate learning rate of stochastic gradient descent problem of minimizing the loss function, using dropouts for preventing overfitting, Using an optimal number of LSTM cells and layers. The detailed proposed algorithm for hybrid course recommendation is depicted as Algorithm 2.

5. IMPLEMENTATION AND PERFORMANCE EVALUATION

The entire system is programmed in Python on a Jupyter notebook with an i-7 9th generation CPU and an Nvidia GeForce RTX 2080 Ti GPU. For preprocessing the data, Tokenization is done using

Algorithm 1.
Modified spectral clustering

```
Input: Similarity matrix $S \in R^{n\times n}$, number k of clusters to construct
Output: Clusters $A_1, . . ., Ak$ with $A_i = \{j| y_j \in C_i\}$.

Begin
Step 1: Compute the similarity graph using the entropy of each number rather than distance or similarity between two points.
The weighted adjacency matrix is denoted by W.
Step 2: Calculate the normalized Laplacian $L_{sym}$.
Step 3: Find the first k eigenvectors $v_1, . . ., v_k$ of $L_{sym}$.
Step 4: Consider the matrix $V \in R^{n\times k}$ which has the vectors $v_1, . . ., v_k$ as columns.
Step 5: From V, Compute the matrix $U \in R^{n\times k}$ by normalizing the row sums to have norm 1, that is $u_{ij} = v_{ij}/(\Sigma_k v^2_{ik})^{1/2}$.
Step 6: let $y_i \in R_k$ be the vector corresponding to the i-th row of U for i = 1, . . ., n.
Step 7: Using K means method, group the points $(y_i)_{i=1,, n}$ into clusters $C_1, . . ., C_k$.
Step 8: Compute the accuracy of the cluster using the Tang Index.
End
```

Algorithm 2.
Proposed hybrid course recommendation algorithm

```
Input: User input Query q
Output: Sequence of recommendation as output.

Begin
    Step 1: Preprocess the data removing unnecessary words. This step includes Tokenization, Lemmatization and Stemming.
    Step 2: Get Input sequence for feeding to Long Short-term memory (2.1 and 2.2 are performed parallelly).
            1.1. Build a semantic network of user queries and search history. This will act as an input in the next step for LSTM.
            1.2. Group similar users using Spectral Clustering.
                    1.2.1. Build a sequence of action of similar users which can be fed to LSTM.
    Step 3: Fed the data in 2.1 and 2.2 to LSTM while adding more bias towards 2.1
            1.3. Set specific hyperparameters for maximizing the accuracy of result.
End
```

customized blank space tokenizer, and lemmatization and stemming are done using Porter's stemming algorithm. Construction of Semantic network of users queries was built using ASKNET. For grouping the similar users, Spectral clustering has been employed as an algorithm with slight modification. It is implemented from scratch using Python Libraries like NumPy, Pandas, ScikitLearn and Matplotlib so that the amendment can be incorporated in it.Finally, LSTM was implemented using TensorFlow, and the two inputs were passed to LSTM as parameters to generate the desired output, i.e. recommended course.

5.1 Baseline Models

With our dataset, we compare our model to the following baselines:

1. **PCRS (Gulzar et al., 2018):** An N-gram query classification and expansion-based knowledge discovery system with ontology support for course recommendations.
2. **FRSSP (Gulzar et al., 2018):** A hybrid course recommender system that uses domain Ontology to provide a knowledge model.
3. **SBCRS (Sankhe et al., 2020):** A skill-based Course recommender system that uses c-means and fuzzy clustering.
4. **CUDCF (Huang et al., n.d.):** A score prediction algorithm using cross-user-domain collaborative filtering.

The comparison of our model with other course recommender systems is shown in Table 1. As evident, our model was able to outperform all the baseline models with a huge margin of in all the metrics. We were able to achieve a huge increase of 9.11% increase in precision value compared to the CUDCF which is the second-best performing model in our environment. This shows that our overall model was effective in extracting sequence of personalized course recommendation using LSTM. We achieved a recall and acccuracy value of 96.69% and 95.12% respectively which are 7.78% and 8.81% more compared to the CUDCF model.

PCRS uses localized static ontologies alone with just a Euclidean distance scheme. Though it incorporates the N-Gram approach, the knowledge captured by PCRS is quite sparse. FRSSP uses Domain Knowledge and is a hybridized model but there is no frequent learning involved and lacks dynamic knowledge making the recommendations not very relevant, on the other hand, our model uses dynamic knowledge from real-world knowledge bases like the wiki data and the RDF store which composes knowledge from other real-world e-learning and course recommendation platforms. Metadata integration, as well as Wikidata, increases knowledge density and verifies the correctness of facts.

SBCRS has just a few parameters and it is based on skills though it uses C-means and fuzzy clustering it fails to capture the knowledge from the surrounding environment, on the other hand, our model uses spectral clustering for grouping similar users for a personalized recommendation. CUCDF model focuses more on the user and is not query centered. The entire focus is on collaborative filtering and the cross-user domain makes it effective, but the knowledge captured by the system is again sparse leading to recommendations which deviate whereas our model has a content-based approach too that makes it easier to expand to a big number of users, can capture a user's individual preferences, and can propose niche things to a small number of other people.

The dataset of various user's history and clicks were collected using surveying 75 different users form various domains like Computer Science, Finance, Healthcare and Arts of which 20 were from Computer science, 18 were from Arts, 16 were from Finance and 21 from Healthcare domain as shown in Table 2. For grouping the similar users, we used spectral clustering while the history of various users is used to build a Semantic Network using ASKNET which is then passed to LSTM as an input for prediction.

As shown in Figure 2 and Figure 3, our system outperformed all the baseline papers. Despite the number of recommendations, the proposed approach has the highest F-measure and lowest FDR values. For F-measure, our model was ahead of the best performing baseline models by a huge margin

Table 1.
Experimental results of course recommender system and comparison with the previous models

	Precision	Recall	Accuracy	F1 Measure	FDR
Proposed Model	93.43	96.69	95.12	95.03	0.06
PCRS	80.92	83.69	81.97	82.28	0.19
FRSSP	82.99	86.47	84.79	84.69	0.17
SBCRS	77.79	80.12	78.47	78.93	0.22
CUDCF	84.32	88.91	86.31	86.55	0.15

Figure 2. F-Measure % vs. No. of recommendations

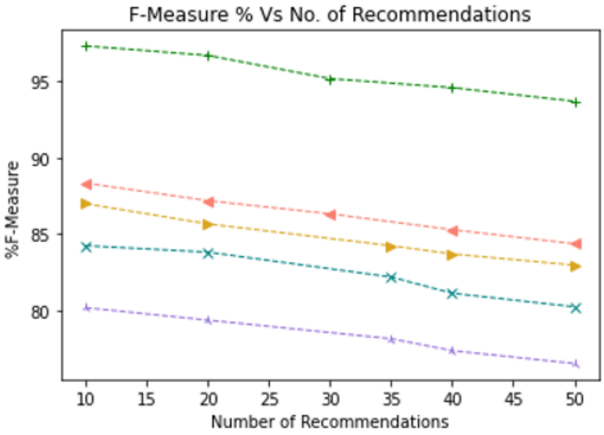
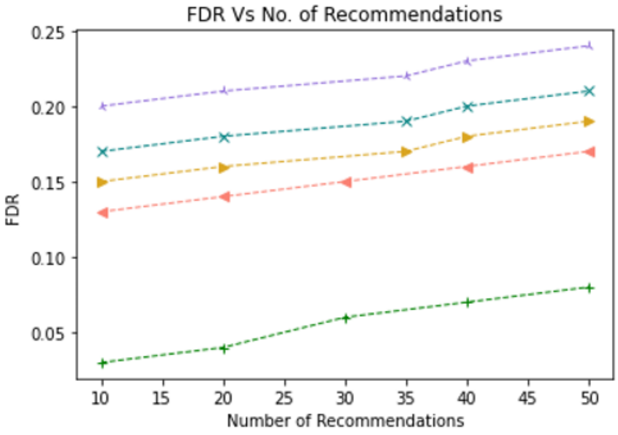


Figure 3. FDR vs. No. of recommendations



of 9% and in the case of FDR, this margin is 0.1. The reason for this high performance is the fact that we are considering the collective user intelligence by spectral clustering of similar users and including the auxiliary knowledge for wiki data and formulating semantic network based on current user click and user queries and also usage of LSTM for classification. This is the approach where both clustering and classification are gauged together to form cumulative successive and independent tasks. Also, we are using a collaborative as well as content-based recommendation into consideration and captures the student's interest successfully. On the other hand, the baseline models have either used collaborative filtering, fuzzy means clustering or have no frequent learning involved and lack dynamic knowledge.

The output for "Machine" gives us a varied list of courses like Machine learning, Machine Learning Python, Machine Learning Finance but all the fields related to the Computer Science (Data Science) domain. So, it shows the recommendations show variation but not out of the domain, which can be useful for the students. In case of "Science" keyword, a lot of variety can be observed, which includes courses from many domains which shows how good the recommendation system works. A similar pattern is maintained for other domain keywords too. Since it takes input from users of similar clusters too and not just content-based filtering, this also works for new users who have no search history and since it takes a content-based approach, it gives highly personalized results. A list of all the results yielded for various search queries are shown in Table 3.

Performance of the recommendations can be evaluated by four terminologies, Recall, Accuracy Precision and F-measure. As demonstrated in Equation 3, precision is defined as the number of relevant courses retrieved divided by the total number of courses extracted. Recall is a related word that is defined as the number of retrieved and relevant courses divided by the entire number of relevant courses for the user, as indicated in Equation 4. The average of precision and recall is defined as accuracy, Equation 5, whereas F-measure, Equation 6, is the harmonic mean of the two variables. Table 4 provides all four metrics for a variety of queries. Precision, recall, accuracy, and F-Measure

Table 2.
Information on all the courses with their domain

Domain Details	No. of Courses
Computer Science	110
Data Science	38
History	45
Finance	78
Health And Medical	130

Table 3.
The results obtained for different search queries

Search Query	Returned Result
Machine	Machine Learning, Machine Learning Python, Machine Learning Stanford, Machine Learning Finance, Data Science: Machine learning
History	History of Finance, History of Psychology, History of Artificial Intelligence, History of China
Finance	Finance Marketing, Investment Management, Business and Financial Modeling, Financial Management
Healthcare	The business of healthcare, Healthcare marketplace, Healthcare information literacy for data analytics, Healthcare organization Operations
Science	Data Science, Applied Data Science, The Science of Well-Being, Methods and Statistics in Social Science, Science of Happiness

numbers indicate how well the answers compare to a query that is relevant to the student's interests. The greater the value of these parameters, the better the system is.

$$\text{Precision} = \frac{No.of\,relevant\,courses\,retrieved}{Total\,no.of\,courses\,retrieved}, \tag{3}$$

$$\text{Recall} = \frac{No.of\,relevant\,courses\,retrieved}{Total\,no.of\,relevant\,course}, \tag{4}$$

$$\text{Accuracy} = \frac{Precision + Recall}{2}, \tag{5}$$

$$\text{F-measure} = \frac{2 * Precision * Recall}{Precision + Recall}, \tag{6}$$

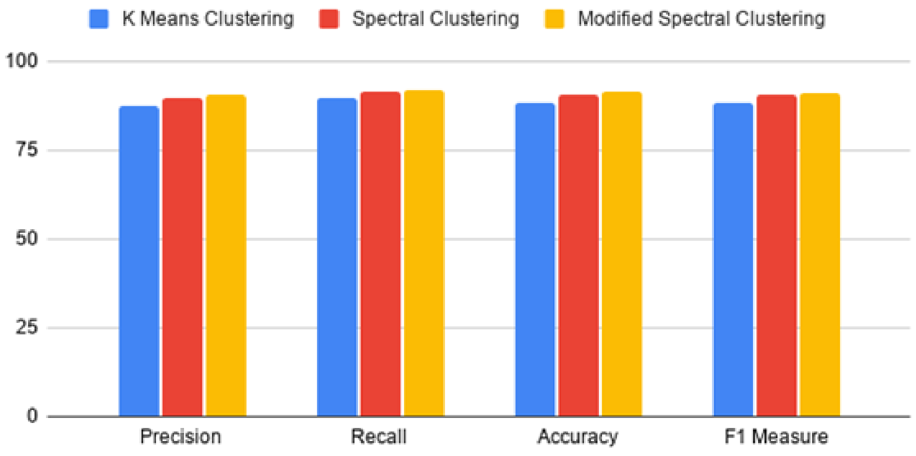
$$\text{FDR} = 1\text{- Precision} \tag{7}$$

When compared with the metrics of other clustering algorithms incorporated in the same system, our modified spectral clustering outshined different clustering algorithms as shown in Fig 4. It performed much better than K-means algorithm, which is the most common clustering algorithm by a significant margin. In every metric, the average difference between K-means and our spectral clustering algorithm is 3%. Compared to Simple Spectral clustering too, our algorithm performed a

Table 4.
Performance analysis of the hybrid course recommender system

Query	Precision%	Recall%	Accuracy%	F-Measure%
Machine	90.54	91.45	90.99	90.99
History	91.25	92.96	92.10	92.09
Finance	90.97	91.72	91.34	91.34
Healthcare	91.45	93.26	92.36	92.35
Science	89.61	91.09	90.35	90.34

Figure 4.
Performance comparison when different clustering algorithms are employed



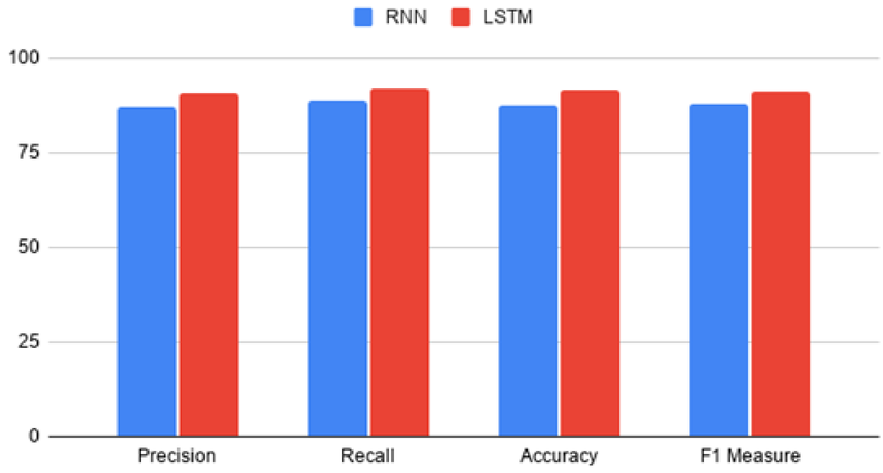
little better every time. The average difference is nearly 1% which is also significant. The reason for this difference is because we added the Tang Index for getting the accuracy of clustering and selecting the cluster with max Tang index. The use of entropy instead of just similarity also contributes to this significant difference. (Can use Shannon and Renyi Entropies but it will increase the work)

When compared the results with various Recurrent Neural Networks too and results were not so shocking that Long Short-term memory outshined vanilla RNN by a significant margin of nearly 4% as shown in Figure 5. The reason for this huge gap is that RNNs are not able to capture long term dependencies since it's gradient of the loss function decays exponentially with time, thus leading to the vanishing gradient problem. On the other hand, LSTM uses gates to control the vanishing gradient problem and has a memory cell for controlling the flow of information to be retained.

6. CONCLUSION

The paper proposed a hybrid course recommender system for suggesting courses to the subject. It considers personalization as a priority and aims to increase the commercial success of the platform. It achieves it by using a content-based approach of capturing user's preference by building a semantic network that is later fed to LSTM. The collaborative filtering approach is made parallelly by making a sequence of similar users, then constructing a series of actions of related users, which is then fed to LSTM. This way, it achieves personalization as well as works fine with new users too, which are the major drawbacks of individual content-based and collaborative filtering recommender systems. It does it work with a decent accuracy of 91.4 percent, which demonstrates the adequacy and accuracy of the proposed model.Better performance could still be achieved by using more efficient RNN architectures than LSTM, such as GRU, and using attention, as well as adding bidirectionality to RNN, particularly at the objective function level, but the fact that standard LSTM works so well already is yet another proof of its ability to tackle general problems. Future study might focus on employing more permutations of hybridizations to build a collective intelligence of methods that can be utilised to improve accuracy and application response time. The proposed work requires several rounds of ground truth validation by prospective users for validation of results which was one of the major tasks which was a limitation but was later resolved by collecting necessary ground truths.

Figure 5.
Performance comparison of the proposed approach for RNN and LSTM



REFERENCES

Al-Badarenah, A., & Alsakran, J. (2016). An automated recommender system for course selection. *International Journal of Advanced Computer Science and Applications*, *7*(3), 166–175. doi:10.14569/IJACSA.2016.070323

Asadi, S., Jafari, S., & Shokrollahi, Z. (2019). Developing a Course Recommender by Combining Clustering and Fuzzy Association Rules. *Journal of Artificial Intelligence and Data Mining*, *7*(2), 249–262.

Bhattacharya, M., Gupta, D., Biswas, S., & Bansal, G. (2018). Designing Recommender System for Corporate Education WiP+ 32. In 2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI) (pp. 72-76). IEEE. doi:10.1109/SmartWorld.2018.00047

Bhumichitr, K., Channarukul, S., Saejiem, N., Jiamthapthaksin, R., & Nongpong, K. (2017). Recommender Systems for university elective course recommendation. In *2017 14th International Joint Conference on Computer Science and Software Engineering (JCSSE)* (pp. 1-5). IEEE. doi:10.1109/JCSSE.2017.8025933

Bobadilla, J., Ortega, F., Gutiérrez, A., & Alonso, S. (2020). Classification-based Deep Neural Network Architecture for Collaborative Filtering Recommender Systems. *International Journal of Interactive Multimedia & Artificial Intelligence*, *6*(1), 68. doi:10.9781/ijimai.2020.02.006

Buhagiar, N., Zahir, B., & Abhari, A. (2018). Using Deep Learning to Recommend Discussion Threads to Users in an Online Forum. In *2018 International Joint Conference on Neural Networks (IJCNN)* (pp. 1-8). IEEE. doi:10.1109/IJCNN.2018.8489194

Chen, W., Cai, F., Chen, H., & De Rijke, M. (2019). Joint Neural Collaborative Filtering for Recommender Systems. *ACM Transactions on Information Systems*, *37*(4), 1–30. doi:10.1145/3343117

Esteban, A., Zafra, A., & Romero, C. (2018). *A Hybrid Multi-Criteria Approach Using a Genetic Algorithm for Recommending Courses to University Students*. International Educational Data Mining Society.

Gruver, N., Malik, A., Capoor, B., Piech, C., Stevens, M. L., & Paepcke, A. (2019). Using Latent Variable Models to Observe Academic Pathways. arXiv:1905.13383. doi:10.4018/IJWLTT.2018070104

Gulzar, Z., & Anny Leema, A. (2018). A framework for recommender system to support personalization in an e-learning system. *International Journal of Web-Based Learning and Teaching Technologies*, *13*(3), 51–68. doi:10.4018/IJWLTT.2018070104

Gulzar, Z., Leema, A. A., & Deepak, G. (2018). Pcrs: Personalized course recommender system based on hybrid approach. *Procedia Computer Science*, *125*, 518–524. doi:10.1016/j.procs.2017.12.067

Hong, W., Zheng, N., Xiong, Z., & Hu, Z. (2020). A Parallel Deep Neural Network Using Reviews and Item Metadata for Cross-domain Recommendation. *IEEE Access : Practical Innovations, Open Solutions*, *8*, 41774–41783. doi:10.1109/ACCESS.2020.2977123

Hu, Q., & Rangwala, H. (2019). Academic Performance Estimation with Attention-based Graph Convolutional Networks. arXiv preprint arXiv:2001.00632. doi:10.1109/ACCESS.2019.2897979

Huang, L., Wang, C.-D., Chao, H.-Y., Lai, J.-H., & Yu Philip, S. (2019). A score prediction approach for optional course recommendation via cross-user-domain collaborative filtering. *IEEE Access : Practical Innovations, Open Solutions*, *7*, 19550–19563. doi:10.1109/ACCESS.2019.2897979

Jiang, D., Liu, Z., Zheng, L., & Chen, J. (2020). Factorization meets neural networks: A scalable and efficient recommender for solving the new user problem. *IEEE Access : Practical Innovations, Open Solutions*, *8*, 18350–18361. doi:10.1109/ACCESS.2020.2968297

Moubayed, A., Injadat, M., Shami, A., & Lutfiyya, H. (2020). Student Engagement Level in e-Learning Environment: Clustering Using K-means. *American Journal of Distance Education*, *34*(2), 1–20. doi:10.1080/08923647.2020.1696140

Nassar, N., Jafar, A., & Rahhal, Y. (2020). A novel deep multi-criteria collaborative filtering model for recommendation system. *Knowledge-Based Systems*, *187*, 104811. doi:10.1016/j.knosys.2019.06.019

O'Donovan, J., & Smyth, B. (2005). Trust in recommender systems. *Proceedings of the 10th international conference on Intelligent user interfaces*, 167-174.

Ognjanovic, I., Gasevic, D., & Dawson, S. (2016). Using institutional data to predict student course selections in higher education. *The Internet and Higher Education*, *29*, 49–62. doi:10.1016/j.iheduc.2015.12.002

Polyzou, , Athanasios, & Karypis. (2019). Scholars Walk: A Markov Chain Framework for Course Recommendation. *Proceedings of the 12th International Conference on Educational Data Mining*, 396-401.

Pornwattanavichai, A., Jirachanchaisiri, P., Kitsupapaisan, J., & Maneeroj, S. (2020). Enhanced Tweet Hybrid Recommender System Using Unsupervised Topic Modeling and Matrix Factorization-Based Neural Network. In *Supervised and Unsupervised Learning for Data Science* (pp. 121–143). Springer. doi:10.1007/978-3-030-22475-2_7

Sankhe, V., Shah, J., Paranjape, T., & Shankarmani, R. (2020). Skill Based Course Recommendation System. In *2020 IEEE International Conference on Computing, Power and Communication Technologies (GUCON)* (pp. 573-576). IEEE.

Sultana, J., Rani, M. U., & Farquad, M. A. H. (2019). Knowledge Discovery from Recommender Systems using Deep Learning. In *2019 International Conference on Smart Systems and Inventive Technology (ICSSIT)* (pp. 1074-1078). IEEE. doi:10.1109/ICSSIT46314.2019.8987766

Zhu, H., Paschalidis, I. C., Chang, A., Stern, C. E., & Hasselmo, M. E. (2020). A neural circuit model for a contextual association task inspired by recommender systems. *Hippocampus*, *30*(4), 384–395. doi:10.1002/hipo.23194 PMID:32057161