



Dynamic Educational Recommender System Based on Improved Recurrent Neural Networks Using Attention Technique

Hadis Ahmadian Yazdi, Seyyed Javad Seyyed Mahdavi Chabok & Maryam Kheirabadi

To cite this article: Hadis Ahmadian Yazdi, Seyyed Javad Seyyed Mahdavi Chabok & Maryam Kheirabadi (2022) Dynamic Educational Recommender System Based on Improved Recurrent Neural Networks Using Attention Technique, Applied Artificial Intelligence, 36:1, 2005298, DOI: [10.1080/08839514.2021.2005298](https://doi.org/10.1080/08839514.2021.2005298)

To link to this article: <https://doi.org/10.1080/08839514.2021.2005298>



© 2021 The Author(s). Published with license by Taylor & Francis Group, LLC.



Published online: 11 Dec 2021.



Submit your article to this journal [↗](#)



Article views: 846



View related articles [↗](#)



View Crossmark data [↗](#)

Dynamic Educational Recommender System Based on Improved Recurrent Neural Networks Using Attention Technique

Hadis Ahmadian Yazdi^a, Seyyed Javad Seyyed Mahdavi Chabok^b,
and Maryam Kheirabadi^a

^aDepartment of Computer Engineering, Neyshabur Branch, Islamic Azad University, Neyshabur, Iran;

^bDepartment of Electrical Engineering, Mashhad Branch, Islamic Azad University, Mashhad, Iran

ABSTRACT

Most web-based educational systems contain some drawbacks, as compared to traditional classrooms. Particularly, it becomes difficult for teachers to guide students to choose an appropriate learning resource due to the large number of online learning resources. Meanwhile, student decisions make it more difficult to choose educational resources according to their circumstances. In this matter, the resource recommender system can be employed as an educational environment to recommend the educational resource advice for students, so that these recommendations can be coordinated to each student's preferences and needs. This paper presents the resource recommender system as a combination of MLP, BiLSTM, and LSTM improved deep learning networks using the attention method. Compared to similar studies conducted using DBN networks and focus only on the near past interests and preferences of users, the proposed system provides higher accuracy and more appropriate recommendations considering current interests, in addition to the user's long-term past interests. The proposed recommender system with accuracy of 0.96 and a loss of 0.0822 contains a better performance to recommend resources to students compared to other methods.

ARTICLE HISTORY

Received 30 March 2021

Revised 3 November 2021

Accepted 8 November 2021

Introduction

Many years ago, the book was well known as the primary tool for education. Today's, however, the development of computers and the World Wide Web and the increase in heterogeneous information has created a sense of the need to design systems to generate the most meaningful recommendations, which simplifies selection and activity processes. Recommender systems have evolved over the past two decades (Adomavicius and Tuzhilin 2005; Cremonesi et al. 2011). By becoming the Internet to a comprehensive medium and the rapid growth of e-learning, users' expectations of these systems have

CONTACT Seyyed Javad Seyyed Mahdavi Chabok  Mahdavi@mshdiau.ac.ir  Department of Electrical Engineering, Mashhad Branch, Islamic Azad University, Mashhad, Iran

© 2021 The Author(s). Published with license by Taylor & Francis Group, LLC.

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

risen. In this regard, one of the advantages of these systems is the lack of space and time constraints on educating. Traditional education systems are time-consuming, as compared to modern education (Shishehchi et al. 2011).

Nevertheless, web-based systems also have some significant drawbacks, as compared to traditional classrooms. For instance, there is no interaction between the learner and the educator. Meanwhile, the presentation of content and feedback is not personalized (Romero, Ventura, and García 2008). Besides, a large number of resources in e-learning environments makes it difficult to make the proper choice. Another point is that the student has individual differences such as educational background, study method, age, etc. This requires us to get feedback from students to better guide them in the educating process (Lu 2004). Concerning such limitations, students in e-learning systems are eager for personalized services to monitor, support automatically, and evaluate student learning. Student loyalty also increases with personalized service (Huimin, Ming, and Mingming 2010; Muthukumar and Bhalaji 2020).

So far, counseling systems, especially educational counseling, have been studied with different techniques and methods. Most of these researches, especially in the issue of educational recommenders, have tried to solve the problem with linear forms and models and data mining such as ontology to recommend scientific resources. (Bourkougou and Achbarou 2018; Bourkougou and El Bachari 2018; Gulzar, Leema, and Deepak 2018; Qiao and Hu 2018; Yago et al. 2018) One of the problems and limitations of these methods is not accepting a large amount of information. Only a framework for recommendation is introduced; there has not been much discussion about a training advisor who automatically offers helpful advice. One of the newest and most complete ways to solve the problem is to use deep learning networks. Due to a large amount of problem data, neural network integration, learning techniques, and structural dynamics informing several hidden layers, these networks can solve problems with high accuracy and not encounter overfit errors. (Alashkar et al. 2017; Chen et al. 2017b, 2017a; Hu et al. 2018; Wang et al. 2018, 2017; Xu et al. 2017; Zheng, Noroozi, and Ph 2017) On the other hand, in issues such as our problem that we are faced with a variety of areas and parameters influencing decision-making, Deep learning networks are more powerful than traditional methods.

Educational Recommender Systems (ERS) is utilized to assist students and teachers during the learning process. Here, it is worth mentioning that the main difference between ERS and their business counterparts is the appropriate educational principles for both the learning and teaching process. This difference in the educational methods used in different academic conditions determines the primary guidelines for the ERS design. Based on this analysis, along with educating and upgrading of existing algorithms, five specific areas are introduced, in which the future research and development can be expected: construction of universal ERS, ERS intended primarily for teachers, ERS that

links student achievements across different courses, ERS which take into account physical distance between students and use of ERS to motivate students to work continuously. (Muthukumar and Bhalaji 2020; Zhang et al. 2018) The use of recommenders when analyzing facts in the decision-making process is one of the basic elements that many people apply during this process (Prem and Vikas 2010). The development and improvement of existing systems are one of the current researches in the world. These developments are applied based on the continuous evolution of statistical methods, machine learning, artificial intelligence, data mining, and information retrieval (Prem and Vikas 2010; Manouselis et al. 2012). This paper aims to design an Educational Recommender System (ERS) that recommends the relevant resources to users based on their interests and features in the relevant dataset. Indeed, our recommender system contains individual features such as educational background, age, and so on, including the individual's interests in pre-clicked, the downloaded resources, and the user score given to each resource. Hence, such a system must be educated to be able to recommend new resources to users.

if there is an inherent structure that the model can exploit, deep neural networks are very efficient for this issue. For instance, both CNN and RNN have long employed the internal structure of machine vision or natural language. Because the nature of recommending textbooks depends on the time and long-term review of student performance, the sequential structure of sessions or report clicks is very appropriate for inferential errors in conventional or recursive models. In many of these methods, the same weight is considered in learning for all users' interests, and only the user's past information is used in learning. While in the present article, having a network that looks both backward and forwards can also cover changes in learner behavior and offer more up-to-date recommendations.

In the following, the sources and descriptions are reviewed on the basic methods on which ERS operates. Then, some examples of different ERS classifications are provided according to their specific characteristics and basic methods. In the next section, the proposed method is presented, which is a combination of the architecture of MLP deep learning networks. In the fourth section, the results of the implementation of the proposed algorithm based on accuracy and efficiency are surveyed. Ultimately, some future suggestions are provided in the fifth section.

Recommender Systems: A Review

In recommender systems, we encounter a set of users, options, and their transactions in the system. Options are entities that are recommended to users based on their content and user transaction with the system. These recommendations are related to various decision-making processes, such as buying an entity, listening to music, or reading online news. A recommender

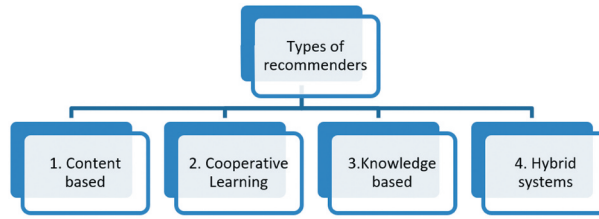


Figure 1. Types of recommender systems.

system usually focuses on a specific type of entity. For example, we can refer to an article or news item to provide valuable and practical recommendations for that particular type of entity. In recent years, research on recommender systems has increased compared to other information systems and methods (such as datasets with search engines) (Covington, Adams, and Sargin. 2016). Nowadays, many recommender systems are running, which are based on different approaches and methods. The methods are shown in Figure 1.

Deep Learning-based Methods

Currently, deep learning has revolutionized the structure of recommenders. It has attracted the attention of many researchers by coping with many of the barriers to traditional models and generating some quality recommendations. Deep learning can receive non-linear user-item relationships and display abstract representations of data at higher layers. Moreover, it can extract complex relationships within conceptual, textual, and visual data (Zhang, Yao, and Sun 2017). An example of a beautiful feature of neural networks and deep learning is that they are end-to-end differentiable and provide suitable inductive biases for the type of input data. As such, deep neural networks can combine several neural building blocks into a differentiable function and educate end-to-end. Here, the key advantage is when it comes to content-based recommendation systems. Multi-modal data is very common for user-item modeling on the web. For instance, when working with textual data such as review data (Zheng, Noroozi, and Yu2017), tweets (Gong and Zhang 2016), items, image data (social posts, product images), CNN/RNNs become the main building blocks. Here, less attention has been paid to the traditional solutions such as modality-specific features, and as a result, the recommender system cannot take advantage of video learning (Zhang et al. 2018).

Educational Recommender Systems (ERS)

Educational recommender systems are increasingly utilized as tools to assist students and teachers in implementing the learning process (Muthukumar and Bhalaji 2020; Zhang et al. 2018). Here, e-learning is one of the fields that its use is

inevitable to improve the quality of education. E-learning is a form of education provided using various electronic tools (Internet, intranet, extranet, satellite networks, audio and videotapes, CDs). It is controlled in different ways (self-directed/controlled by the educator), and its implementation is without geographical and time restrictions (simultaneous/asynchronous learning). Other terms are used to describe this method of learning and educating, such as online learning, virtual learning, distributed education, and web-based learning (Mubarak, Cao, and Ahmed 2021; Tejeda-Lorente et al. 2015).

The Most Important Challenges in ERS

Despite the many improvements that recommendation systems have made, these systems also encounter some challenges that can be summarized as follows: Automatic information retrieve, cold start problem (user/new items), highly specialized content, the lack of diversity in recommendations, data scarcity problem, fraud problem, and critical mass problem (Zhang et al. 2018).

In the Automatic information retrieval problem, the main issue is that today's algorithms have limited ability to analyze the content of recommended items automatically. Items with associated textual content (such as books, web pages, etc.) are usually easily described (using different approaches for Information retrieval from texts). The most developed algorithms are tailored for analyzing textual content (Santos and Boticario 2011). They use keywords and phrases that are found in the text and compare them with search parameters. With the higher correlation between these data, the likelihood that a particular text can be recommended to the designated user is greater (Adomavicius and Tuzhilin 2005).

The Cold-start (New User/Item) problem appears in situations when ERS encounters a user or an item that could be recommended for the first time. In such cases, the system does not have enough information about this user or the item to prepare a meaningful recommendation (Al Mamunur, Karypis, and Riedl 2008). Consequently, the system depends on the manually entered initial parameters about the user or the items of recommendation provided by the user or system administrator. Implicitly gathered information about the user, which does not require the user's cooperation, will give the system more accurate information about the user's interest, how the user uses the design, what contents are recommended, etc. (Reddy, 2016). However, for implicit data collection, the user must use the system for a certain period of time. In open education surroundings there is a danger that, due to the lack of information about the new items, they will not be treated like that by the system. In these cases, ERS relies only on the available information about items that are in some cases dependent on the other users of the plan (through ratings, etc.). (Adomavicius and Tuzhilin 2005; Al Mamunur, Karypis, and Riedl 2008).

Content overspecialization and non-diversity problem are pronounced when the ERS only recommends items that score highly with the user's profile. In these cases, there is a risk that the user will only be recommended very similar items. In ERS, this issue is more pronounced in open educational environments that usually determine recommendations based on matching the user's profile and recommended items (Sunil and Saini 2013). In formal education environments, teachers could rectify the system to ensure that various items are recommended (in accordance with the objectives of the course). On the other hand, in open education environments, the most common approach for solving this problem is the introduction of a random selection of content that will be recommended, taking into account that there is a proper correlation between this content and the content the user is interested in (Adomavicius and Tuzhilin 2005; Cremonesi et al. 2011).

The Sparsity and Gray Sheep problem usually appears when ERS depends on the ratings of items by the system's users or when the recommendation is done based on grouping and comparing similar users. Suppose some items that the system can recommend have been evaluated by a few users of these items, regardless of their quality. In that case, they will not be widely recommended to other users. In addition to the items' content, the problem of sparsity could appear among system users. The user who does not fit well in any of the groups will not get good recommendations.

In formal educational environments, these problems can be solved through interventions done by teachers. However, in open education environments, there is a risk that they will remain unresolved (Adomavicius and Tuzhilin 2005).

Fraud problem in ERS is related to the data entered by the user. These data could be basic data on/about the user's profile or the data collected through tests used for monitoring user advancement through the course. Although fraud problems make no sense in open education environments, in formal education environments where achievement in an assignment may have consequences for the overall success of the user, there is a possibility of fraud. This can happen when the user is not monitored during the use of the ERS.

Examples of Educational Recommender Systems

Today, many different ERSs are utilized. Their objective is to facilitate the modernization of the educational process in both formal and open educational environments. These systems are usually a combination of design and behavior, methods, and strategies to create the recommendations. In this way, ERS can be divided into systems that recommend educating, learning objects, teamwork to implement joint activities, different educating methods through learning cases based on the user's unique preferences, or helping to create

a personal learning path (PLP) (Zhang et al. 2018). Likewise, the ERS educating method recommends dividing users into methods in formal learning environments and freely available methods on the World Wide Web.

Regarding the widespread use of web2.0 tools for e-learning, most ERSs recommend a combination of these methods. Besides, some ERSs help teachers to perform a part of student supervision (Tejeda-Lorente et al. 2015) or find some ways to recommend the learning topics (Huang, Zeng, and Chen 2007; Mubarak, Cao, and Ahmed 2021). Sunita et al. (Aher and Lobo 2012) recommend ERS courses available to students. Then, they developed their recommender system based on the best combination of lessons available and each user's unique interests. In (Imran et al. 2016) developed their recommender system of Personalized Learning Object Recommender System (PLORS), which is an ERS in the LMS by various learning objects to personalize the formal educational process, based on monitoring the activities of previous students and then comparing them with other students and their activities. In (El-Bishouty et al. 2014), they proposed an ERS model that helps teachers proportion e-learning content to their students' different learning styles.

Moreover, the E-learning Activities Recommender System (ELARS) (Hoic-Bozic, HolenkoDlab, and Mornar 2016) exploits visual, auditory, read/write, mobile (VARK) descriptions (Fleming 1995), and learning styles as an essential element in a user profile. (Marian, Popescu, and Costel 2015) suggest the use of ERS to help students find the groups who can help them solve a particular problem in learning the content of a specific course. In several different systems, the use of ERS to communicate with students has anonymously been proposed.

In addition, one of the objectives of ELARS is to be able to recommend when forming a group to work on a specific problem or on a particular project. When this feature is added to the ERS, students usually have the freedom to decide independently whether to accept recommendations and communicate with appropriate colleagues or to ignore them (Zhang et al. 2018). Determining a personal learning path is one of the objectives of a number of ERSs. These systems employ different input parameters to define a unique path of educational content for each user. China Ming et al. (Chin Ming, Chih Ming, and Mei Hui 2005) arranged the syllabus in such a way that the system utilizes the student's incorrect answers to devise more learning paths so that the user can obtain sufficient knowledge of the course content.

On the other hand, Lata et al. (Latha and Kirubakaran 2013) developed a model of ERS that its algorithm employed the graph theory and knowledge of different learning styles to recommend different PLPs for each user. In (Chin Ming et al. 2007), each user's basic knowledge level is compared to the complexity of individual learning objects. According to the results of this comparison, the ERS provides some recommendations for different learning paths. Besides,

in (Onah and Sinclair 2015), the PLP building was designed based on a comparison of the user profile and the aim of proper learning defined by the user. The ERS monitors the user's progress and changes the learning path to ensure that all the required knowledge is acquired to succeed in further learning.

To obtain the better performance of the algorithms used in ERS, several methods of artificial intelligence (fuzzy sets, artificial neural networks, evolutionary strategies) or their interactions are utilized (Zhang et al. 2018). In the papers of (Tejeda-Lorente et al. 2015; Jamsandekar and Mudholkar 2013), they use fuzzy inference methods to process data on student success with the aim of better monitoring students' progress through course content. Artificial neural networks are considered to develop algorithms that are capable of self-learning based on data from a given domain (Negnevitsky 2005), on the ERS of artificial neural networks for complex modeling relationships between user profiles and their interests (De Gemmis et al. 2009), as well as for modeling the relationship between recommended objects and other parameters in the ERS to determine specific recommendations that are unique for each user (Adomavicius and Tuzhilin 2005; Jamsandekar and Mudholkar 2013). This method can obtain better overall results in the same environment, as compared to cases where only one of these methods is employed (Zhang et al. 2018). Some methods of artificial intelligence based on evolutionary calculations include genetic algorithms, evolutionary strategies, and genetic programming (Negnevitsky 2005). It should be noted that genetic algorithms and various evolutionary strategies are commonly used in ERS (Zhang et al. 2018). In (Sengupta, Sahu, and Dasgupta 2011), they used the Ant Colony Optimization approach (an evolutionary algorithm) to identify system users' effective and optimal learning paths. This system is exploited to access information about unfamiliar terms that the user encounters during the learning process. The paper (Chin Ming et al. 2007) utilized the genetic algorithm to create a personal learning path for the user, while (Cayzer and Aickelin 2002) used the biological immune system model to obtain a set of possible recommendations.

Proposed Method

This paper aims first to obtain a dataset of users, including their interest in the resources under study and the extent to which they use and click on these resources and related features. After that, the practical items from them are chosen. In the second phase, using deep neural networks, we educate our recommender system with acceptable accuracy, and finally, we recommend resources to users using the educational network. The recommender algorithm encompasses data extraction from OULAD data source files, data preprocessing, construction of a deep learning network of MLP, LSTM, and BiLSTM networks improved by Attention method, initialization of parameters, educating, and finally, educating point predicting. In our proposed architecture, as illustrated in Figure 2, in each layer for each feature in the

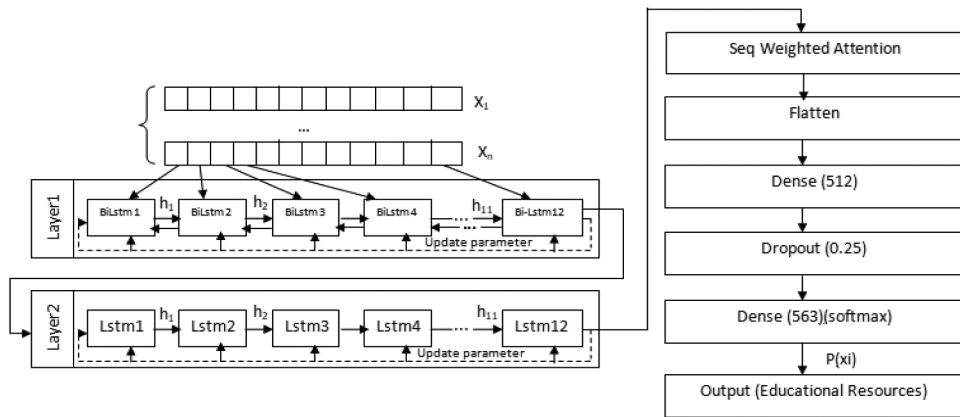


Figure 2. Proposed network architecture.

dataset, one BiLSTM cell and one LSTM cell are considered so that the cells are focused on one feature of each record. Each cell considers only the pattern of one feature. Finally, the total of these patterns leads to obtaining a better result.

Predicting Resource Scores

In the process of model educating, data labeled as class is considered as an educating set to model educating. Then, according to the user-lesson attribute vector, the recommendation problem becomes the category prediction problem. In this paper, using the label of rating classes, the loss information is published to each layer from top to bottom by accurately tuning the observer's parameters. After educating the model to obtain a certain amount of loss, a set test can be employed to test the performance of the recommender model. The data in the set test is categorized into two classes: user-lesson attribute vector and lesson assessment. Each user-lesson attribute vector corresponds to a category level, while each level corresponds to a score. All lessons that correspond to a user are sorted based on the expected scores, and then the lesson recommendations are generated.

Using the Bilstm structure in the first layer due to its two-sided nature focuses on short-term and long-term interests. In this architecture, two layers of LSTM and Bilstm are siblings used to extract the general patterns in the total database data. Finally, the output of these two layers is sent to the attention layer. In this architecture, we have used the Seq weighted model of the attention technique to reduce the useless features and side effects of noise data; At the beginning of the proposed architecture classification step, the Dense layer was used, and since the inputs of these layers are vectors, we used the Flatten layer to convert the output of the higher layers into vectors. As you can see in [Figure 2](#), before the last layer, which is the softmax output and

determines the probability of belonging to each class, dropout has been used to use the secret aggregation, ensemble feature; Also, it prevents network overload and increases the generalization capability of the model.

Implementation

The volume of dataset data used is about 11 million data, which after eliminating those records via missing fields, finally remains 10543682 records consisting of 12 features. It should be noted that this set of records is gathered from the activities of 23326 different students. The implementation steps are carried out in two parts: the first part contains the hybrid architecture (see [Figure 2](#)) that refers to our idea in the paper. On the other hand, the second part includes implementing several traditional methods and the deep learning based on the DBN network related to the idea of the study of (Zhang et al. 2018), which developed scientific resources. In this article, we have used 3 divisions 70% – 30%, 80% – 20%, 90% – 10% to train and test our proposed model, and for each case 3 values of 0.1, 0.2, 0.3 For validation split. (Table 7) After the completion of the Epochs, the graphs and the results of the performances show that the accuracy and loss of our work are far better than the results of the implementation of the proposed network (Zhang et al. 2018; Lic. Mar'ia Emilia Charnelli et al. 2019; Hui Chen et al. 2020; Rumei Li et al. 2019; Lingyao Yan et al. 2021)

Dataset

Students generate a diversity of behavioral data by learning in an online learning environment. This behavioral data is gathered and stored through data collection methods (Kuzilek, Hlostá, and Zdrahal 2017). The resources dataset provides data sources for this platform. This resources dataset can be exploited to extract content features that reflect students' interest in resources. The feature vectors of students are generated by combining student features and lesson features. Afterward, the combination vectors of behavioral feature and user-lesson feature are created. The dataset consists of 3 students, teachers, and lessons. It includes information about 22 courses provided, 32,593 students, their assessment results, and their interaction with the Virtual Learning Environment (VLE), which is summarized by Students' daily clicks on various "resources" (10,655,280 inputs) are provided. The dataset is anonymous using the ARX data encryption tool [PK15]. The data is investigated for loss detection and then confirmed and published by the Open Data Institute¹.

OULAD is a sample subset of the collected student data, which contains student demographics, student performance in the course assessment, and last but not least, student behavior in the VLE. This resource provides a unique dataset of student performance and prepares an opportunity to create new

generations of learning management systems. Courses (called modules) with a history of at least two successive presentations were chosen as the first stage. This course covers the topic of learning and the set of sessions completed with the test.

Module-presentation: represents the academic year so that the courses are taught. After that, the data is converted and identified using a data Anonymizer tool.

Dataset Plan

Figure 3 depicts the overall structure of the presented data set. This data set generally represents students and then the course. The main table of StudentInfo contains the student files that are linked to the courses (A student can have more than a one registered course). Each course has several assessments related to the student using the student assessment table, including the history of the student assessment results.

- **courses.csv:** The file contains a list of all available and presented courses.
- **assessments.csv:** This file contains information about the assessments of the presented courses. Usually, each course has a number of assessments followed by a final exam.

There are three types of assessment: Teacher-Made Assessment (TMA), Computer-Made Assessment (CMA), and Final Exam (Exam).

Data Description

- **vle.csv:** Contains information about the tool's existence in the VLE. It is usually HTML, pdf, etc., pages. Students have access to these resources online, and their interactions with them are then recorded.
- **studentInfo.csv:** This information includes demographic information about students and their results. Each student may have several rows. Each row contains information about one student in the studied course, including the following columns:
- **studentRegistration.csv:** This table encompasses information about the student registration time to participate in the module. Besides, it is recorded for students who have not registered the registration date.
- **StudentAssessment.csv:** This file contains student assessment results. If the student does not submit an assessment, no result is recorded. Meanwhile, if the assessments are not stored in the system, there will be no final exams.
- **studentVle.csv:** it includes information about each student's interaction with resources in the VLE.

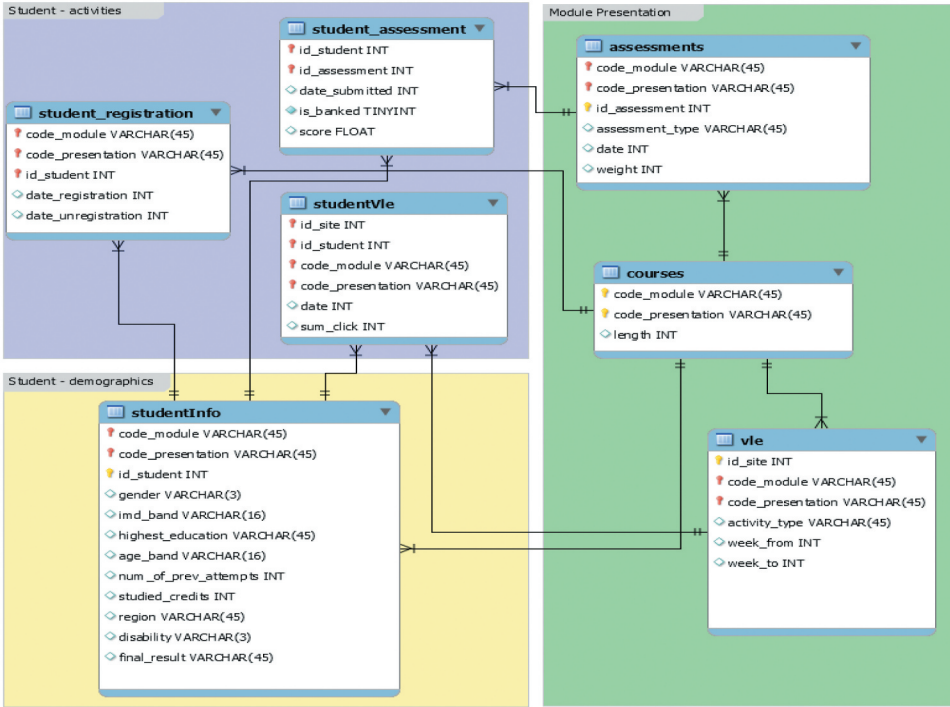


Figure 3. Dataset structure of the data set (Kuzilek, Hlosta, and Zdrahal 2017).

Data Pre-processing

Figure 4 shows the pre-processing steps. The input consists of four sections: provided resources, student features, held courses, and student performance and assessment history in each course. These four sections are combined to perform further analysis, categorization, feature mapping. Table 1 clearing blank or mistake data, and feature normalization.

The features are normalized before implementing any work (the data range is [0, 1]). To do so, Eq. (1) is utilized to perform the normalization as follows:

$$x^* = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (1)$$

Where Xmin denotes the lowest eigenvalue, which is $Xmin = \min\{X1, X2, \dots, Xn\}$. Xmax denotes the maximum eigenvalue that is $Xmax = \max\{X1, X2, \dots, Xn\}$. X^* indicates the normalized value, x means the primary data. Ultimately, after pre-processing, the data is divided into an education set and a test set.

Records Labeling

The dataset is labeled after the initial stages of pre-processing as follows:

For each student from the set of registered activities for the joint courses, the resource with the most clicks (which can indicate the student's taste and interest) in the course that had the highest grade (which can be the influence of the

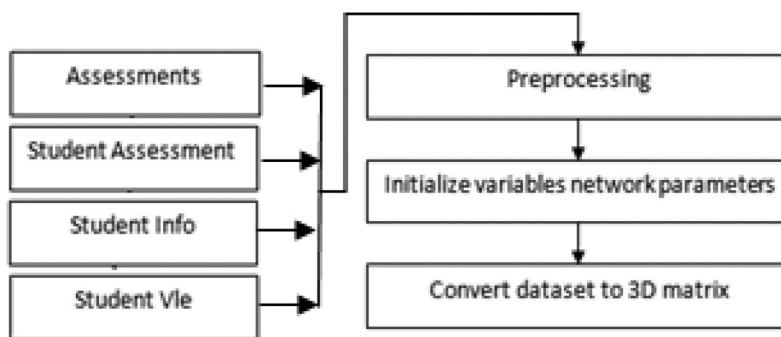


Figure 4. Data pre-processing steps.

resources studied in that course to show the student to be more successful) was chosen as the label. Therefore, 562 labels were generated, which were mapped from 0 to 561. The frequency of repetition of labels is illustrated in [Figure 5](#).

In the end, after the pre-processing step, the data is divided into an education set, a validation set, and a test set. The education set enters our proposed network as input.

Network Construction

The current research was conducted on a Google Colab server. Here, it is worthwhile to mention that Google Colab is a cloud service provided by Google that allows Python programming, which prepares to install and work with several Python language packages and deep learning frameworks such as Tensorflow, Keras Pytorch, and more. In terms of this service, it provides a free GPU to users, which has practically multiplied the efficiency of this service. The service has been provided with an Nvidia Tesla P100 and 25.51GB of RAM. Besides, both LSTM and Bidirectional Keras library have been used to build the recommended network. After entering data into the network, the data enters the Bidirectional layer with 1536 neurons. At the end of the output of this layer, they enter the LSTM layer with 512 neurons, and at the end of the output of this layer, they enter the Attention layer. In this implementation, SeqWeightedAttention existence in the Keras library has been employed to implement the attention technique.

Initialization of network parameters:

In the model educating process, we must repeatedly adjust the model parameters to achieve better results in feature extraction. During the learning process, the minibatch process method is exploited to solve the problem of large data volumes. Besides, some parameters such as learning rate, number of repetitions, and Batch-Size are set as follows:

Batch-size: 2048, Learning rate: 0.0001, Epoch: 120, Activation function: Softmax

Table 1. Values of mapped features to numbers.

code_module	AAA = 0.1	BBB = 0.2	CCC = 0.3	DDD = 0.4	EEE = 0.5	FFF = 0.6	GGG = 0.7
code_presentation	2013B = 540	2013 J = 720	2014B = 180	2014 J = 360	-	-	-
age_band	0-35 = 0.1	35-55 = 0.2	55 ≤ = 0.3	-	-	-	-
highest_education	A Level or Equivalent = 0.1	HE Qualification = 0.2	Lower Than A Level = 0.3	No Formal quals = 0.4	Post Graduate Qualification = 0.5	-	-
final_result	Distinction = 0.1	Fail = 0.2	Pass = 0.3	Withdrawn = 0.4	-	-	-
gender	F = 0.1	M = 0.2	-	-	-	-	-

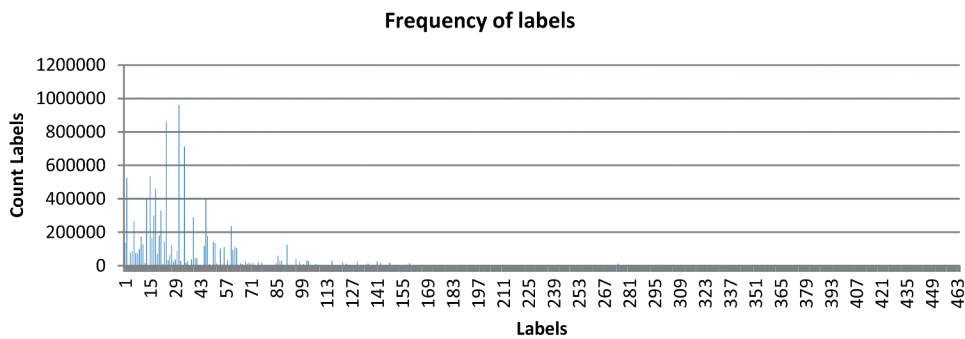


Figure 5. Frequency of repetition labels.

The higher the number of network parameters, the higher its computational load in the network educating phase. In our proposed network, in the educating phase, the first val accuracy can be observed = 0.95 at epoch = 74 via a minimum loss of 0.08 at epoch = 115. After completing the educating phase, we entered the test data as input to the network, and then the obtained final result was equal to 0.96 with one percentage of generalization.

Methods and Tools of Data Analysis

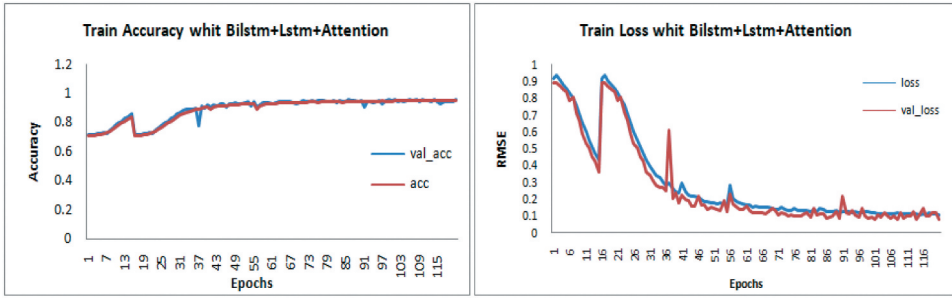
The recommender system developed in this paper aims to predict the best sequence of educational resources. To do so, there are many criteria to measure different aspects of recommending performance. Two essential criteria, i.e., Mean Absolute Loss (MAE) and Root Mean Squared Loss (RMSE), are utilized to measure the accuracy of the predicted scores (P) for each educational resource relative to the correct scores (R).

Here, to assess the loss of the implemented methods, RMSE is employed. Then both the accuracy and loss of our architecture are compared to ones of the architecture of the study (Zhang et al. 2018), which contains a similar application nature to our work (suggested by scientific sources). Table 7 and Figure 6 exhibit the accuracy and loss of our network in the educating phase.

Result and Discussion

Investigate the Effect of the Number of Cells in Each Layer

As can be observed in Table 3, the results of single-cell structures in three single-layer architecture, two layers, and three layers, are implemented and then investigated, which are not desirable. An LSTM cell could not find well the pattern of different features, the relationship of features to themselves, or other features in combination, permutation, and additional models.



b. The accuracy diagram in our proposed network at 120 epochs a. The loss diagram in our proposed network at 120 epochs

Figure 6. Results of educating and testing of our proposed network(test-split = 0.3, split-valid = 0.1).

As shown in Table 4, the single-layer multicellular architecture implemented in the single-layer two architectures of LSTM, the single-layer BiLSTM was implemented and then investigated and obtained more favorable results.

Investigating the Effect of Using the Attention Mechanism

As mentioned before, the attention technique can filter out useless features from raw inputs and decrease the side effects of noise data. By applying the attention technique to the recommender systems, we can eliminate useless content and choose the items via the most representation along with maintaining interpretability. We added the attention technique to single-layer two architectures of LSTM and BiLSTM. As can be observed in Table 5, the use of the attention technique in network architecture has had a positive effect on the obtained results.

Investigating the Effect of the Number of Layers on Neural Network Structure

In deep learning network architecture, the first problem is the relationship of LSTM cells to themselves, which was investigated in detail. The second problem is the relationship between the different layers in the implemented architecture

Table 3. Results of educate and testing of LSTM networks (n, 1, 12).

accuracy	test	val_ accuracy	accuracy	val_loss	loss	train	1layer
0.28		0.2879	0.2825	2.6617	2.6907		
accuracy	test	val_ accuracy	accuracy	val_loss	loss	train	2layer
0.18		0.1792	0.2607	3.3326	2.8305		
accuracy	test	val_ accuracy	accuracy	val_loss	loss	train	3layer
0.25		0.2548	0.266	2.8369	2.8633		

Table 4. Results of educating and testing of LSTM and BiLSTM networks (n, 12, 1).

accuracy	test	val_ accuracy	accuracy	val_loss	loss	train	LSTM
0.79		0.7876	0.7696	0.5544	0.6194		
accuracy	test	val_ accuracy	accuracy	val_loss	loss	train	BiLSTM
0.70		0.7086	0.6868	1.0078	1.1137		

Table 5. Results of educating and testing of LSTM and BiLSTM network (n, 12, 1) and the attention technique.

accuracy	test	val_ accuracy	accuracy	val_loss	loss	train	LSTM & Attention Techniques
0.94		0.9366	0.9423	0.1254	0.1457		
accuracy	test	val_ accuracy	accuracy	val_loss	loss	train	BiLSTM & Attention Techniques
0.70		0.758	0.7458	0.6675	0.72223		

that this part of the architecture performs the general understanding of the features in the database. Here, to survey the effectiveness of the relationship between the layers, we implemented and educated two-layer two architectures LSTM and two-layer Gru. As shown in Table 6, the results of two-layer architectures are more favorable than those of single-layer architectures.

As the BiLSTM deep neural network considers both long-term and short-term interests of the user, and due to their gradual learning natures, they support learners’ behavioral changes. Hence, we implemented our proposed two-layer architecture as a combination of the improved LSTM and BiLSTM network using the attention technique. As can be seen in Table 7, the results of our proposed architecture are very acceptable and desirable

According to Figure 6 considering the loss rate, it can be seen that the number of selected epics is appropriate. Besides, according to the accuracy diagram, it can be seen that as the val and val accuracy diagrams are almost the same, so that overfitting does not occur in this experiment.

Investigating the Effect of Unconventional Data on Model Accuracy

As shown in Figure 5, we are faced with an unbalanced data set. In these cases, in addition to the model’s accuracy, it is better to use other evaluation parameters such as recall and F1 score. To ensure that the model has not

Table 6. Results of educating and testing of two-layer LSTM and GRU networks.

accuracy	test	val_ accuracy	accuracy	val_loss	loss	train	LSTM
0.93		0.9265	0.911	0.1622	0.198		
accuracy	test	val_ accuracy	accuracy	val_loss	loss	train	GRU
0.94		0.9205	0.9066	0.1806	0.2175		

Table 7. The result of proposed network in terms of train and test accuracy.

test accuracy	train					Split-valid	Test-split
	val_ accuracy	accuracy	val_loss	loss			
0.9664	0.9597	0.9511	0.0822	0.1055	0.1		30
0.9674	0.9564	0.9473	0.0909	0.1134	0.2		
0.965	0.9581	0.949	0.087	0.1123	0.3		
0.9684	0.965	0.9576	0.0704	0.0902	0.1		20
0.9593	0.9516	0.9438	0.102	0.1236	0.2		
0.9631	0.9577	0.9484	0.0863	0.1125	0.3		
0.964	0.9568	0.9574	0.088	0.1025	0.1		10
0.9611	0.9574	0.9468	0.0878	0.1145	0.2		
0.9603	0.9567	0.9462	0.091	0.1175	0.3		

intelligently categorized all the data presented into a repetitive class in the training process to achieve high accuracy. At the end of the test phase, for all three parameters precision-recall f1-score, the average value is 96%. We examined the classification accuracy of each group separately and found that the data of all classes were very well categorized. To evaluate, we have trained and tested our model with 3 different data sharing modes. In [Table 8](#), as an example, we have ten cases of the groups that had the lowest frequency of repetition and ten cases of the groups that had the highest frequency of repetition from the test results.

As you can see, in all three divisions of 30–70, 20–80 and 10–90 in the groups with the value support = 1, the result of most recalls and f1-scores is equal to 1. On the other hand, the number 1 obtained for recalls and f1-score is very small in the groups with the most members. This shows that our model has also met the unconventional data challenge in addition to the data volume challenge.

Comparison of the Performance of the Proposed Model with Other Models

We have compared the suggested model results in the first row of [Table 8](#) with other methods presented in related work or implemented by ourselves. As can be seen, the results are more desirable for different evaluation parameters of the proposed model than other implemented methods. All evaluations were performed on OULAD shared data.

The proposed method (Zhang et al. 2018) has been implemented and trained, tested, and evaluated with OULAD data. As shown in [Table 9](#), it performed worse than our proposed model in terms of both error and accuracy criteria. Five methods have been implemented and studied in (Lic. Mar'ia Emilia Charnelli et al., 2019). The results show that the SVD algorithm with an error of 0.839 is more desirable than other schemes, which is several times higher than the error of our proposed model.

In (Hui Chen et al. 2020), the three criteria, including Recall, Prec, and F1 for the three methods itemCF, Clustering + itemCF, and AROLS are examined. It shows that the proposed algorithm (AROLS) has a better Prec than the other two cases. Meanwhile, F1 and Recalls remain relatively steady at n top recommendation at the same time.

The work (Rumei Li et al. 2019) shows that AROLS performs much better than traditional participatory filtering, especially User-AROLS calling and accuracy, which has more than tripled. Also, the calling accuracy of UserCF is much smaller than ItemCF, probably because UserCF focuses more on the interests of learners who are more like a particular learner. At the same time, the ItemCF recommendation is more personal because it primarily suggests similar items based on the learner's interest. As can be seen in the first row, the proposed model performed better than all seven reviewed methods.

Table 8. Values of mapping properties given in numbers.

Train test split(test-size = 0.1)					Train test split(test-size = 0.2)					Train test split(test-size = 0.3)				
support	f1-score	recall	precision	label	support	f1-score	recall	precision	label	support	f1-score	recall	precision	label
1	0.4	1	0.25	530	1	0.4	1	0.25	209	1	1	1	1	326
1	1	1	1	498	1	1	1	1	228	1	1	1	1	418
1	1	1	1	453	1	1	1	1	547	1	1	1	1	463
2	0.02	0.5	0.01	477	1	0.4	1	0.25	228	1	0.4	1	0.25	391
2	1	1	1	305	1	1	1	1	90	2	1	1	1	446
2	1	1	1	460	2	1	1	1	184	2	0.4	1	0.25	442
3	1	1	1	213	3	1	1	1	133	2	0.01	0.5	0.01	171
3	1	1	1	503	3	1	1	1	391	3	1	1	1	408
3	0.02	0.5	0.01	411	3	1	1	1	326	3	1	1	1	547
3	1	1	1	103	4	1	1	1	557	3	1	1	1	550
82212	0.96	0.96	0.96	8	82212	0.96	0.96	0.97	1	86317	0.95	0.95	0.96	16
84858	0.99	0.99	0.99	34	84858	0.98	0.98	0.98	57	94754	0.98	0.98	0.99	20
99036	0.99	1	0.98	5	99036	0.98	0.99	0.98	60	112033	0.98	0.99	0.98	12
101120	0.98	0.99	0.98	28	101120	0.96	0.96	0.97	5	112418	0.99	1	0.99	46
111173	0.96	0.95	0.98	56	111173	0.97	0.97	0.98	11	131070	0.97	0.97	0.98	17
137559	0.98	0.99	0.98	13	137559	0.98	0.99	0.98	18	148366	0.98	0.99	0.98	1
165625	0.97	0.98	0.97	33	165625	0.98	0.98	0.99	16	154820	0.98	0.98	0.99	14
198928	0.98	0.98	0.99	49	198928	0.98	0.98	0.99	19	202925	0.98	0.98	0.98	34
262277	0.97	0.97	0.97	56	262277	0.96	0.96	0.97	26	249276	0.97	0.96	0.97	23
265671	0.98	0.98	0.98	12	265671	0.95	0.95	0.96	41	275379	0.98	0.97	0.98	30

10 groups with the highest frequency

10 groups with the lowest frequency

Table 9. Comparison of the proposed method with other methods.

Model	Accuracy	RMSE	Recall	Prec.	F1	ref
proposed model	0.959	0.08	-	-	-	Implemented by us
train						
test	0.96	-	0.96	0.96	0.96	
Naive Bayes(nb)	0.1981	-	0.4725	0.2937	0.1853	
Logistic Regression(Lr)	0	-	0	0	0	
Latent Dirichlet Allocation (lda)	0.5415	-	0.0772	0.4314	0.4579	
DBN	0.2912	0.225	-	-	-	(Zhang et al. 2018)
KNN Baseline (item-item)	-	0.915	-	-	-	(Lic. Mar'ia Emilia Charnelli et al.2019)
KNN Baseline (user-user)	-	0.908	-	-	-	
KNN Means (item-item)	-	0.898	-	-	-	
KNN Means (user-user)	-	0.944	-	-	-	
SVD	-	0.839	-	-	-	
AROLS	-	-	0.022	0.28	0.04	(Hui Chen et al. 2020)
itemCF	-	-	0.018	0.18	0.027	
Clustering + itemCF	-	-	0.024	0.24	0.041	
ItemCF	-	-	0.026	0.1334	0.0435	(Rumei Li et al. 2019)
Item-AROLS	-	-	0.0406	0.1880	0.0668	
User-AROLS	-	-	0.0018	0.0046	0.0026	
UserCF	-	-	0.0005	0.0011	0.0007	
CF with ARM	-	-	0.6874	0.076	0.1374	(Lingyao Yan et al. 2021)
The proposed method	-	-	0.8647	0.1033	0.1842	

In (Lingyao Yan et al. 2021), the results show that OLS characters can make the recommendation algorithm more accurate and robust, but as seen in the first row, the proposed model performed better than both studied methods.

Conclusion and Future Works

Today's ERSs do not usually include the designed mechanisms to assist teachers in decreasing their workload. Such systems are more focused on the needs of students. Regarding the work proposed by (Bhojak, Jain, and Muralidharan 2012), some internal algorithms are designed to assist teachers. Overall, the data collected by the ERS can noticeably be employed to help teachers (Zhang et al. 2018). Because of this diversity, the developed systems for one environment may not efficiently be utilized (without a significant change in how they work) in a different learning environment. Currently, the employed systems are specialized for one of these two learning environments. Indeed, a range of future research and development will construct ERSs that can work appropriately with minimal changes in both environments. By introducing the Bologna process in higher education, particularly in continuous monitoring and evaluation of student work, the amount of teacher work has significantly increased (Zhang et al. 2018). Regarding the lack of functional understanding, one area of further ERS research and development will certainly focus on teacher support, especially in formal learning environments.

Systems should take teachers' workload completely, especially when we need the continuous monitoring and evaluation of student work during the semester. Although in terms of education, some algorithms developed for the ERS and

investigated in one course could be employed without any modification in another course (algorithms are not dependent on the content education). Generally, systems do not create any link in the student achievement to different courses. In fact, considering that the education programs based on learning outcomes and obtaining the general and specific qualifications are predefined, the obtained results in one course can be considered to provide recommendations in another course (Zhang et al. 2018). The differences existed between the different study methods that are suitable for use in other areas, requiring the system's flexibility to satisfy the needs of all users. Concerning these differences, the ERS model can be designed and built to provide some satisfactory services to the students and teachers who utilize them. The areas for the future development of ERS confirm that there are still numerous opportunities for further scientific advancement in the field of ERS (Zhang et al. 2018).

Disclosure Statement

No potential conflict of interest was reported by the author(s).

ORCID

Seyyed Javad Seyyed Mahdavi Chabok  <http://orcid.org/0000-0003-1856-4579>

References

- Adomavicius, G., and A. Tuzhilin. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering* 17 (6):734–49. doi:10.1109/TKDE.2005.99.
- Aher, S. B., and L. M. R. J. Lobo. 2012. Course recommender system in E-learning. *International Journal of Computer Science and Communication* 3 (1):159–64.
- Al Mamunur, R., G. Karypis, and J. Riedl. 2008. Learning preferences of new users in recommender systems: An information theoretic approach. *ACM SIGKDD Explorations Newsletter Journal* 10 (2):90–100. doi:10.1145/1540276.1540302.
- Alashkar, T., S. Jiang, S. Wang, and Y. Fu (2017, February). Examples-rules guided deep neural network for makeup recommendation. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 31, No. 1).
- Bernhard, T. J. (1997). Challenges and strategies for electrical engineering education. In 27th Annual Conference Frontiers in Education: Teaching and Learning in an Era of Change (pp. 1459–1462). Pittsburgh, Pennsylvania, USA. doi:10.1109/FIE.1997.632708.
- Bhojak, H., S. Jain, and V. Muralidharan (2012). Instructor guided personalization of learningpath-adopting SCORM. In 4th International Congress on Engineering Education – ImprovingEngineering Education: Towards Sustainable Development. New York, USA. doi:10.1109/ICEED.2012.6779280.
- Bourkhoukou, O., and E. El Bachari. 2018. Toward a hybrid recommender system for e-learning personalization based on data mining techniques. *JOIV: International Journal on Informatics Visualization* 2 (4):271–78. doi:10.30630/joiv.2.4.158.

- Bourkhoukou, O., and O. Achbarou. 2018. Weighting based approach for learning resources recommendations. *JOIV: International Journal on Informatics Visualization* 2 (3):104–09. doi:[10.30630/joiv.2.3.124](https://doi.org/10.30630/joiv.2.3.124).
- Cayzer, S., and U. Aickelin (2002). A Recommender System based on the Immune Network. In Congress on Evolutionary Computation CEC2002, (pp. 807–813), Honolulu, USA. doi:[10.1109/CEC.2002.1007029](https://doi.org/10.1109/CEC.2002.1007029).
- Chen, C., P. Zhao, L. Li, J. Zhou, X. Li, and M. Qiu (2017a, April). Locally connected deep learning framework for industrial-scale recommender systems. In Proceedings of the 26th international conference on World Wide Web companion (pp. 769–770).
- Chen, J., H. Zhang, X. He, L. Nie, W. Liu, and T. S. Chua (2017b, August). Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention. In Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval (pp. 335–344).
- Chin Ming, H., C. Chih Ming, and C. Mei Hui (2005). Personalized Learning Path Generation Approach for Web-based Learning. In 4th WSEAS International Conference on E-activities (pp. 62–68). Miami, Florida, USA.
- Chin Ming, H., C. Chih Ming, C. Mei Hui, and C. Shin Chia (2007). Intelligent web-based tutoring system with personalized learning path guidance. In 7th IEEE International Conference on Advanced Learning Technologies, ICALT 2007 (pp. 512–516). Nigata, Japan.
- Covington, P., J. Adams, and E. Sargin. (2016). Deep neural networks for youtube recommendations. In Recsys. 191–98.
- Cremonesi, P., F. Garzotto, S. Negro, A. V. Papadopoulos, and R. Turrin (2011). Looking for good Recommendations: A comparative Evaluation of Recommender Systems. In P. Campos, N. Graham, J. Jorge, N. Nunes, P. Palanque, and M. Winckler (Eds.), 13th IFIP TC 13 International Conference INTERACT, 2011, (pp. 152–168). Lisbon, Portugal. doi:[10.1007/978-3-642-23765-2_11](https://doi.org/10.1007/978-3-642-23765-2_11).
- De Gemmis, M., Iaquina, L., Lops, P., Musto, C., Narducci, F., and Semeraro, G. (2009). Preference learning in recommender systems. *Preference Learning*, 41, 41–55.
- El-Bishouty, M. M., T. W. Chang, S. Graf, and N. S. Chen. 2014. Smart e-course recommender based on learning styles. *Journal of Computers in Education* 1 (1):99–111. doi:[10.1007/s40692-014-0003-0](https://doi.org/10.1007/s40692-014-0003-0).
- Fleming, N. D. (1995, July). I’m different; not dumb. Modes of presentation (VARK) in the tertiary classroom. In Research and development in higher education, Proceedings of the 1995 Annual Conference of the Higher Education and Research Development Society of Australasia (HERDSA), HERDSA (Vol. 18, pp. 308–313).
- Gallego, D. L., E. Barra, A. Gordillo, and G. Huecas (2013). Enhanced recommendations for e-learning authoring tools based on a proactive context-aware recommender. In IEEE Frontiers in Education Conference (pp. 1393–1395). Oklahoma City, USA. doi:[10.1109/FIE.2013.6685060](https://doi.org/10.1109/FIE.2013.6685060).
- Gong, Y., and Q. Zhang (2016). Hashtag recommendation using attention-based convolutional neural network. In IJCAI, 2782–88.
- Gulzar, Z., A. A. Leema, and G. Deepak. 2018. Pcrs: Personalized course recommender system based on hybrid approach. *Procedia Computer Science* 125:518–24. doi:[10.1016/j.procs.2017.12.067](https://doi.org/10.1016/j.procs.2017.12.067).
- Hidasi, B., A. Karatzoglou, L. Baltrunas, and D. Tikk (2016). Session-based recommendations with recurrent neural networks. In Proceedings of the International Conference on Learning Representations, San Juan, Puerto Rico, 2–4 May 2016, 1–10.
- Hoic-Bozic, N., M. Holenko Dlab, and V. Mornar. 2016. Recommender system and web 2.0 Tools to enhance a blended learning model. *IEEE Transactions on Education* 59 (1):39–44. doi:[10.1109/TE.2015.2427116](https://doi.org/10.1109/TE.2015.2427116).

- Hu, B., C. Shi, W. X. Zhao, and P. S. Yu (2018, July). Leveraging meta-path based context for top-n recommendation with a neural co-attention model. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (pp. 1531–40).
- Huang, Z., D. Zeng, and H. Chen. 2007. A comparison of collaborative-filtering recommendation algorithms for ecommerce. *IEEE Intelligent Systems* 22 (5):5. doi:10.1109/MIS.2007.4338497.
- Huimin, Q., C. Ming, and X. Mingming (2010, May). A personalized resource recommendation system using data mining. In 2010 International Conference on E-Business and E-Government (pp. 5365–5368). IEEE.
- Imran, H., M. Belghis-Zadeh, T. W. Chang, and S. Graf. 2016. PLORS: A personalized learning object recommender system. *Vietnam Journal of Computer Science* 3 (1):3–13. doi:10.1007/s40595-015-0049-6.
- Kuzilek, J., M. Hlosta, and Z. Zdrahal, Open university learning analytics dataset, <https://www.kaggle.com/vjcalling/oulad-open-university-learning-analytics-dataset>
- Latha, C. B. C., and E. Kirubakaran. 2013. Personalized learning path delivery in webbased educational systems using a graph theory based approach. *Journal of American Science* 9 (12):981–92.
- Lu, J., A personalized e-learning material recommender system, Macquarie Scientific Publishing, pp. 374–379, 2004.
- Manouselis, N., H. Drachsler, K. Verbert, and E. Duval. 2012. *Recommender Systems for Learning*. Springer Science & Business Media.
- Marian, C. M., P. S. Popescu, and I. Costel (2015). Intelligent tutor recommender system for on-line educational environments. In 8th International Conference on Educational Data Mining (pp. 516–519). Madrid, Spain.
- Mubarak, A. A., H. Cao, and S. A. Ahmed. 2021. Predictive learning analytics using deep learning model in MOOCs' courses videos. *Education and Information Technologies* 26 (1):371–92. doi:10.1007/s10639-020-10273-6.
- Muthukumar, V., and N. Bhalaji. 2020. MOOCVERSITY-deep learning based dropout prediction in MOOCs over weeks. *Journal of Soft Computing Paradigm (JSCP)* 2 (3):140–52. doi:10.36548/jscp.2020.3.001.
- Negnevitsky, M. 2005. *Artificial intelligence – A Guide to Intelligent Systems*. London: Addison-Wesley, Pearson Education Ltd.
- Onah, D. F., and J. E. Sinclair (2015, March). Massive open online courses: An adaptive learning framework. In Proceedings of the 9th International Conference on Technology, Education and Development (pp. 1258–1266).
- Qiao, C., and X. Hu (2018, July). Discovering student behavior patterns from event logs: Preliminary results on a novel probabilistic latent variable model. In 2018 IEEE 18th International Conference on Advanced Learning Technologies (ICALT) (pp. 207–211). IEEE.
- Romero, C., S. Ventura, and E. García. 2008. Data mining in course management systems: Moodle case study and tutorial. *Computers & Education* 51 (1):368–84. doi:10.1016/j.compedu.2007.05.016.
- Santos, C. O., and G. J. Botcarario. 2011. Requirements for semantic educational recommender systems in formal e-learning scenarios. *Open Access Journal: Algorithms* 4 (1):131–54. doi:10.3390/a4030131.
- Sengupta, S., S. Sahu, and R. Dasgupta. 2011. Construction of learning path using ant colony optimization from a frequent pattern graph. *International Journal of Computer Science* 8 (6):314–21.
- Shishehchi, S., S. Y. Banihashem, N. A. M. Zin, and S. A. M. Noah (2011, June). Review of personalized recommendation techniques for learners in e-learning systems. In 2011 International Conference on Semantic Technology and Information Retrieval (pp. 277–281). IEEE.

- Sunil, L., and D. Saini (2013). Design of a Recommender System for Web Based Learning. In World Congress on Engineering WCE2013 (pp. 3–8). London, England.
- Tejeda-Lorente, A., J. Bernabe-Moreno, C. Porcel, P. Galindo-Moreno, and E. Herrera-Viedma. 2015. A dynamic recommender system as reinforcement for personalized education by a fuzzy linguistic web system. *Elsevier - Procedia ComputerScience* 55 (1):1143–50. doi:10.1016/j.procs.2015.07.084.
- Wang, J., L. Yu, W. Zhang, Y. Gong, Y. Xu, B. Wang, ... D. Zhang (2017, August). Irgan: A minimax game for unifying generative and discriminative information retrieval models. In Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval (pp. 515–24).
- Wang, Q., H. Yin, Z. Hu, D. Lian, H. Wang, and Z. Huang (2018, July). Neural memory streaming recommender networks with adversarial training. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (pp. 2467–75).
- Xu, Z., T. Lukasiewicz, C. Chen, Y. Miao, and X. Meng (2017, August). Tag-aware personalized recommendation using a hybrid deep model. AAAI Press/International Joint Conferences on Artificial Intelligence.
- Yago, H., J. Clemente, D. Rodriguez, and P. Fernandez-de-cordoba. 2018. On-smmle: Ontology network-based student model for multiple learning environments. *Data & Knowledge Engineering* 115:48–67. doi:10.1016/j.datak.2018.02.002.
- Zhang, H., T. Huang, L. V. Zhihan, S. Liu, and H. Yang. (2018). MOOCRC: A highly accurate resource Recommendation model for use in MOOC environments. Springer, Mobile Networks and Applications.
- Zhang, S., L. Yao, and A. Sun (2017). Deep learning based recommender system: A survey and new perspectives. arXiv, arXiv:1707.07435.
- Zheng, L., V. Noroozi, and Y. Ph. (2017). Joint deep modeling of users and items using reviews for recommendation. In WSDM.