A learning styles based recommender system prototype for edX courses

Jyotirmoy Gope
Department of Computer Engineering
National Institute of Technology, Kurukshetra
Haryana - 136119
India
Email: j.gopee@gmail.com

Sanjay Kumar Jain
Department of Computer Engineering
National Institute of Technology, Kurukshetra
Haryana - 136119
India
Email: skj_nith@yahoo.com

***Abstract*—Recommender Systems (RSs) are information filtering systems that gather data about users, to build useful recommendations for them. In todays age of bigdata, RSs have become important tools for finding relevant items from numerous options. RSs for web-based learning assist learners in discovering relevant learning content, according to requirements. Since learners have diverse needs and contexts, recommendations need to be highly personalized. In web-based learning, MOOCs are globally accessible online courses, designed to support learning of a large number of students of different backgrounds. For modelling users, existing MOOC RSs focus on extrinsic properties such as demographics, ratings, whereas a students learning is more dependent on intrinsic qualities like competency, learning styles. In this work, a new RS prototype has been designed, for recommending MOOCs based on requirements and learning styles. This prototype is based on an adapted version of the FSLSM LS model and works exclusively for edX MOOCs.**

***Index Terms*—Recommender systems; Learning Styles; MOOCs**

I. Introduction

Recommender systems (RS) are information filtering systems that gather various kinds of data about user and items, in order to build recommendations that would be useful to them [9]. RSs have become an integral part of everyday lives. From finding artifacts such as movies and books, to taking rudimentary decisions such as which traffic routes to avoid or places to eat, people are quite reliant on recommendations. In today's age of big-data, RSs have become important tools for searching relevant items from a pool of numerous options [15]. Nevertheless, it also helps those who have little knowledge about a kind of item and are initially unsure of where to look. From the service providers' point of view, RSs are important means of increasing revenue and improving service. In research, recommender systems is a well-established field and has been studied extensively [12].

In the education domain, web-based learning, also known as 'online learnin', 'Virtual education', 'Internet-based learning', etc, is an umbrella term for a suite of tools and technologies, which help students, learn online in a flexible environment [9]. RSs in this domain, assists learners in discovering relevant learning content, that suit their style and nature [16]. In the context of learning, it is important to remember that learners have diverse needs and contexts, with different qualities [16]. Thus, recommendations in web-based learning should be personalized to the highest degree, meeting the needs of each learner individually.

Recently in web-based learning, a steady rise has been witnessed in the popularity of Massive Open Online Courses or MOOCs. MOOCs are globally accessible online courses designed using digital learning components to support learning of a large number of students of different backgrounds [25]. Taking a MOOC is analogous to taking a correspondence course using the internet. Its social connectivity features help create a virtual classroom environment for the students [13]. Students can globally attend the courses and access helpful educational content from anywhere at any time [21]. MOOC providers such as Coursera [1], Udacity [2], edX [3], host numerous MOOCs, covering a multitude subjects and affiliated to a host of premier institutions such as IIT Bombay, Harvard and MIT to name a few. Students can search and enroll in a MOOC on the websites of these MOOC providers.

Existing RSs for MOOCs focus on extrinsic properties such as demographics and ratings for modelling users and courses. But the learning process is much more dependent on the intrinsic properties such as competency, intrinsic motivation, etc. This affects the personalization of the recommendations. Learning style is one such property that is simple to model yet essential for knowledgeable learning. Although the learning abilities of students are mainly dependent on their interests, skills and preparation, their LS affect the quality of the knowledge they acquire [4]. LS of a person is defined as a "unique collection of individual skills & preferences that affects how a person perceives, gathers, & processes information" [14]. LSs affect a person's learning and behavior. Learning styles is an extensively studied field, with many researchers having validated its importance [14].

In this work, a new RS prototype has been designed, for recommending MOOCs based on requirements and learning styles (LS) of students. This prototype is based on an adapted

[1]https://www.coursera.org
[2]https://www.udacity.com
[3]https://www.edx.org

version of the FSLSM LS model and works exclusively for edX MOOCs. The reason for using edX courses is two-folds

a) All edX courses are created on open source foundations, which facilitated access to course data for research.
b) Prior to this, many researchers have had also worked using edX courses [24], which laid a guideline for our work.

The main feature of this prototype is a scoring mechanism, which is based on prior work done by [14] and is used to rank the recommendations. In order to evaluate the system, the ranking utility of the recommendations is measured [11] and compared with that of ratings based recommendations.

The rest of the paper is organized as follows

- Section 2 covers the related works and research trends
- Section 3 covers the system design principles and methods
- In section 4, implementation details are given
- In section 5, the experiments performed for evaluation are explained
- Section 6 displays the results generated
- Section 7 concludes the paper.

Throughout the paper, the terms MOOCs and courses have been used interchangeably.

II. Related Work

Recommender systems is a well-established field and has been studied extensively [12]. Some notable works and examples of MOOC RSs can be found in [17,18,19],[22,23].

Learning styles (LS) is an extensively studied field, with many researchers having validated its importance [14]. LS of a person is defined as a ”unique collection of individual skills & preferences that affects how a person perceives, gathers, & processes information” [4],[14]. There are a number of choices for learning style models to interpret learning styles of a person such as [1,2,3,4]. This work is based on the Felder & Silverman learning styles model (FSLSM), which is a renowned and validated LS model [7]. The FSLSM Model defines four dimensions of learning styles, where each dimension comprises of a pair of learning methods (LM). A learner prefers one of these LMs in each dimension. Thus each dimension can be represented as a pair of LMs. The 4 dimensions are:

- Active/ Reflective
- Sensory/Intuitive
- Visual/Verbal
- Sequential/Global

A learner prefers a LM in each dimension and can have mild, moderate, or strong dependency on it, where a mild learner can adapt between the two LMs, whereas a moderate or strong learner faces difficulty to learn without their preferred LMs. The Learning style of a person is expressed as a combination of preferred LMs in each of these dimensions.

A number of methods exist for identifying the LS of a person based on FSLSM. The Index of Learning Styles (ILS) [5] was the first instrument, that was designed for determining learning style of a student based on FSLSM model [5]. It is a 44-item questionnaire based on which a students preferred learning method in each dimension is determined. Ever since this, many more instruments have been developed, for detecting LS of students by means of interactive sessions or automatically by examining their activities. [8], [10], [20], [26].

III. System Design

This prototype is based on an adapted version of the FSLSM LS model and works exclusively with edX courses. In FSLSM model, students' LSs indicate which learning methods (LMs) they prefer. Using this theory, the prototype scans each course to search for learning objects (LOs) which support the preferred LMs of the student and recommend courses with good support. In practice, this is implemented using a scoring mechanism. For each course, lm_scores are calculated corresponding to six LMs of the FSLSM model, by scanning the course structure for relevant LOs. These scores indicate the course support for each of the six LMs. At the time of recommendation, the prototype selects courses which match the students' requirements and calculate a ls_score for each course using the computed lm_scores and based on the students' LS. The ls_score indicates the course support for the students LS.

The lm_scores are calculated using an implemented algorithm that scans the course structure for supporting LOs. During runtime, the prototype calculates the ls_score by aggregating $lm\_scoress$ corresponding to the LMs preferred by the student. Both ls_score and lm_scores are decimal values and range on a scale of 0 to 5. Courses in the recommendation list are arranged in descending order of ls_score. In the following subsections , a technical explanation of the prototype is given, which covers the design details such as the data considerations and working architecture. Along with this, the algorithm used for computing the scores is also explained.

A. Working Data

1) Learning Styles Model: The prototype uses an adapted version of the FSLSM model for working. According to the ILS version of the FSLSM, each learner has a preferred learning method (LM) in each of the four dimensions of the model. Based on the dependency level on the preferred LMs, learners can be classified into three groups; mild, moderate and strong. Mild learners are regarded to be well balanced and can easily switch between LMs. On the contrary, moderate or strong learners face difficulty in learning something, which does not support their preferred method.

To implement the FSLSM model, we redefine the dimensions to include a balanced preference in each dimension, where a balanced preference means that the learner has a mild dependency level and can adapt to both LMs.

In this work, it is assumed that all edX courses support sequential and global learners both, as because within every course, an outline is provided and students can study in any order [4,5]. On the basis of this assumption, the sequential and global learners have not been included in the working model version. The redefined dimensions are

- Active/Balanced/Reflective
- Sensing/Balanced/Intuitive
- Visual/Balanced/Verbal

As per this, now there are 27(3^3) learning styles possible according this version. Table 1 can be used to map the original learning styles to the new styles. The idea for this version is based on work of [10].

TABLE I
MAPPING OF LS FROM ORIGINAL FSLSM MODEL TO ADAPTED VERSION.

Dim	Learner Preferences		
	Original Version		Redefined Version
	LM	*Dependency Level*	*LM*
1	Active	Moderate/Strong	Active
	Active/Reflective	Mild	Balanced
	Reflective	Moderate/Strong	Reflective
2	Sensing	Moderate/Strong	Sensing
	Sensing/Intuitive	Mild	Balanced
	Intuitive	Moderate/Strong	Intuitive
3	Visual	Moderate/Strong	Visual
	Visual/Verbal	Mild	Balanced
	Verbal	Moderate/Strong	Verbal

2) Course Data: The course structure of edX MOOCs can be divided into four structural blocks, namely sections, subsections, units and components [27]. Figure 1 below illustrates their hierarchical relationship [27].

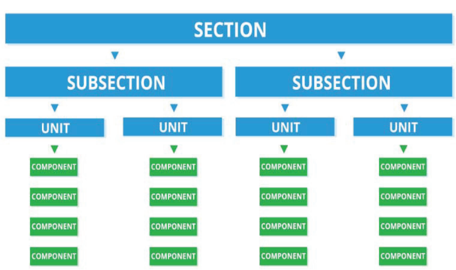

Fig. 1. Hierarchical structuring of edX courses [27]

At edx.org, all the structural metadata of a course are stored in a JSON file [28], as key/value pairs, separated by commas. Curly braces store objects and arrays are denoted using square brackets [28].Within these JSON files, metadata of the structural blocks are stored in the form of JSON objects. This information is sufficient to reconstruct the course structure for scanning LOs and computing the lm_scores.

There are a number of ways to access this JSON file [24], [29]. Using REST APIs [30] is one such way, which allows access to the files via simple web browser. Another advantage is that the JSON files can be downloaded offline, which is helpful for study and experimentation.

B. Architecture

The prototype works, based on the following architecture as shown in figure 2.

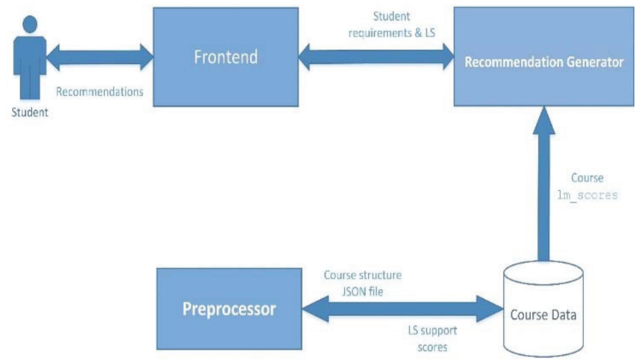

Fig. 2. System architecture based on which prototype is developed

The components in the above architecture are described as follows

(a) *Frontend:* The main purpose of the frontend is to acquire the requirements and learning style of the students, to whom courses are being recommended. The frontend passes this information to the recommendation generator, which uses it to generate personalized recommendations. These recommendations are then displayed to the student using the frontend.

(b) *Recommendation Generator:* Generates a list of recommendations, sorted in order of ls_scores. Firstly, it selects all the courses, which match the students' requirements as the recommendation candidates. It then computes the ls_scores for these courses by aggregating their lm_scores corresponding to the preferred LMs of the student. Final recommendations are provided as a list of courses, sorted in decreasing order of their ls_score.

(c) *Preprocessor:* Computes the lm_scores for the courses and stores it in the database along with the course details. It uses an algorithm to compute the lm_scores vector for each course. A single lm_score for a particular LM is calculated by scanning the course structure for learning objects that support that LM.

C. Algorithm

This algorithm is used to compute the lm_scores of a course corresponding to the six LMs of the working LS model. It has been adapted upon prior work done by [14]. The algorithm accepts the JSON file, containing the structural metadata of the course, as input and scans it for different types of learning objects (LOs). LOs are certain unique course elements, which serve as useful learning tools [14]. For this work, eight types of LOs are used, along with the rules for identifying them from the structural metadata. The basic assumption is that each LM is supported by certain favorable LOs. The lm_score corresponding to a LM is calculated by computing availability and prevalence factors using the frequencies of LOs supporting that LM. The preprocessor uses this algorithm to compute the lm_scores for each course which are stored as vectors along with the course details.

Figure 4 shows the flowchart of the algorithm. The working has been divided into two sub processes, explained next.

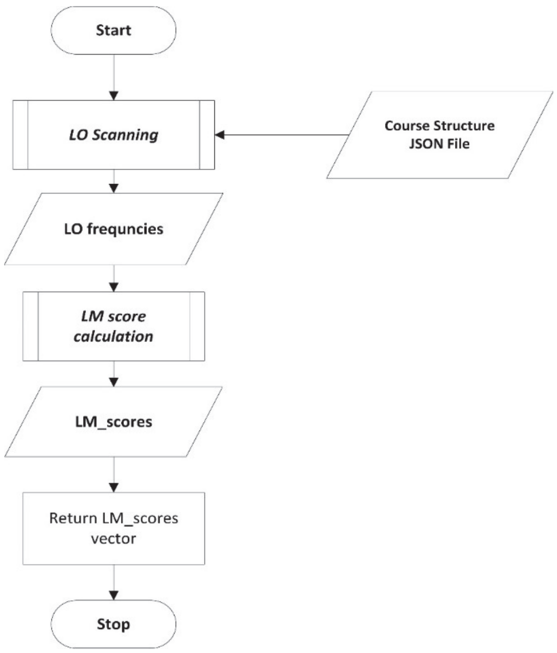

Fig. 3. Flowchart of algorithm

1) LO Scanning: Learning objects (LOs) [14] are uniquely identifiable course content, which act as useful learning tools for students. For this work, the following LOs are used, but more LOs can be added. All the LOs are self-explanatory.

- Exercises
- Quizzes
- Tests
- Applications
- Examples
- Additional Materials
- Videos
- Discussions

This sub process scans the JSON file to find these LOs and count their frequency. The file is parsed to extract blocks of information, storing metadata for a structural elements, such as a section, subsection, etc. Tables 2,3 & 4 give the parsing rules for identifying LO blocks in the JSON file. These tables are based on analysis of the JSON files and information provided in [29].

TABLE II
RULES FOR IDENTIFYING EXERCISE, TEST, QUIZ AND APPLICATIONS

Block Properties		LO
Type	*Format*	
Subsection	activity, assignment, course, homework, learning, lesson, vocabulary	Exercise
	exam, final, midterm, test, problem set	Test
	inline, questions, quick, quiz	Quiz
	understanding, creative, final, lab, programming, project	Applications

TABLE III
RULES FOR IDENTIFYING ADDITIONAL MATERIALS AND EXAMPLES

Block Properties		LO
Type	*Format*	
Units	read, documentation, format, think, dictionary, model, survey, note, examine, reference, research	Additional Material
	example, program, function, cod, project, model, tutorial, lab, method, working, basic, application logic, explore, interview	Examples

TABLE IV
RULES FOR IDENTIFYING VIDEOS AND DISCUSSIONS

Block Properties	LO
Type	
video	Videos
discussion	Discussions

For normalizing the lm_scores, all frequency readings are fit in the following ranges

TABLE V
FREQUENCY RANGE FOR LO TYPES.

LO	Range	LO	Range
Exercise	0 to 10	**Additional Material**	0 to 10
Tests	0 to 4	**Examples**	0 to 24
Quiz	0 to 10	**Videos**	0 to 100
Applications	0 to 10	**Discussions**	0 to 80

2) LM Score calculation: This process generates the lm_scores for the course, corresponding to the six LMs using the frequencies obtained from LO scanning. It is assumed that, each LM is supported by certain LO types. The following table [14] provides a suitability mapping for LOs and LMs.

TABLE VI
SUITABILITY RELATION BETWEEN LO AND LM. A CROSS-MARK DENOTES THAT LO SUPPORTS THE CORRESPONDING LM.

LO/LM	Active	Reflective	Sensing	Intuitive	Visual	Verba
Exercise	x		x			
Test	x		x			
Quiz		x		x		
Applications			x			
Additional Material		x		x		x
Examples		x	x			
Videos	x		x		x	
Discussion	x					x

To calculate the lm_score for a particular LM x; let $suitable$ be the set of LOs that support x and ,let n denote its length. Let $LOFreq[]$ be an array storing the frequencies of each LO type.

To calculate the lm_score for x, two factors are calculated.

(i) *Availability factor:* What fraction of suitable LO types are present in the course.

$$Availability = \frac{\sum_{\forall i \in suitable} min(LOFreq[i], 1)}{n} \quad (1)$$

Availability ranges between 0 & 1.

(ii) *Prevalence factor:* Measures the abundancy of the supporting LOs.

$$Prevalance = \frac{\sum_{\forall i \in suitable}(LOFreq[i]/threshold[i])}{n} \quad (2)$$

Where $threshold[i]$ is minimum frequency value of i^{th} LO. Thresholds are used to define a minimum standard for selecting candidates for recommendations. Threshold for this prototype is set as $3, 1, 3, 3, 3, 6, 25, 20$ for the eight LOs. Prevalence ranges between 0 to 4. lm_score is obtained as the sum of availability and prevalence. Both ls_score and lm_score ranges between 0 to 5.

IV. Implementation

The recommendation generator and preprocessor were implemented using Java language, on 64-bit intel core-i3 processor machine on a Windows based operating system. At boot time, the recommendation generator, calls the preprocessor to compute lm_scores for selected courses and uses it to generate a recommendation list, sorted in order of rating and lm_scores. For data storage, files were used in the backend. The frontend has not been implemented, and is kept open for development. During runtime, the frontend calls the recommendation generator, passing the student requirements and LS as arguments to it. The relevant files and source code are available at https://github.com/Jyogo/RSPrototype

V. Experiments

In general, all RSs compute utility values for items, which indicate the usefulness of the item for the target user [9]. Different RS define and compute this score differently. For this work, the ls_scores can be regarded as the utility values, which are used to order the courses on the final recommendation list. An application of the utility value is to assign ranks to the items. Thus, ranking measures are useful in this case, for evaluation [11].

Evaluation metric. For this work, Average Reciprocal Hit Rank (ARHR) [11],[6] has been used as a metric to evaluate the utility of ls_score against ratings of the courses, given by students who have completed the course. ARHR [11] is an unnormalized measure for measuring utility of recommendations, that assigns a utility $1/k$ to a successful recommendation at position k. If h is the no. of courses that were liked, having rankings $r1, r2, ., rh$, and n is the total no. of courses that were recommended, ARHR is calculated as

$$ARHR = \frac{1}{n}\sum_{i=1}^{h} r_i \quad (3)$$

ARHR scores range between 0 an 1. To evaluate the prototype, the rankings of the courses as per ls_scores are compared with the rankings as per student ratings. In order to do so, an experiment was designed, where:

- 45edX MOOCs were selected for three subjects, out of which 16 courses were for ”algorithms”,16 courses for ”database” and 13 courses for ”machine learning.” JSON files and student ratings were collected for these courses. Ratings for these courses were obtained from the MOOC navigator websites Coursetalk [4] and Class-central [5]. Preprocessing was performed to generate the lm_scores for these courses.
- A user study was conducted where 25 M.Tech students from National Institute of Technology, Kurukshetra participated. For their requirements, the students were asked to choose a subject among the three subjects. LS of the students were recognized using ILS [5].
- Based on the collected data, recommendation lists were generated, in which course links were provided for MOOCs related to the chosen subject along with both rankings, which are based on ls_scores and ratings.
- The students are asked browse the recommendations and answer a feedback survey.

ARHR value was computed based on the feedbacks, for each recommendation made and the average ARHR value was used to infer the results.

VI. Results

For each recommendation, the students were asked to specify which courses they liked. Based on their feedback, the rankings of the liked courses, according to the ls_scores and course ratings, were used for computing ARHR scores for both types of rankings, as per formula (3). Fig 4. graphically depicts the two ARHR scores for each recommendation made. The mean ARHR value is obtained as 0.67666, which means that 67.6% of the recommendations were useful and thus the prototype shows promise.

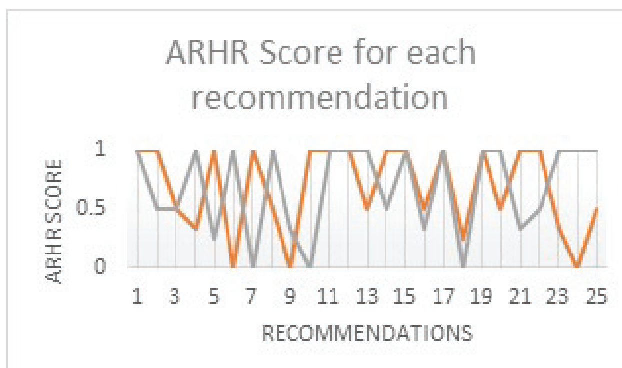

Fig. 4. ARHR Scores for each recommendation for each student

VII. Conclusion

Recommender systems for web-based learning, collects information about students such as their activities, learning preferences, to recommend learning content such as courses, web pages, videos, tutorials, lessons and other learning content, which meet the requirements as well as pedagogical

[4]https://www.coursetalk.com
[5]https://www.class-central.com

characteristics of learners. In the context of learning, it is important to remember that it is a sensitive affair and learners have diverse needs and characteristics, with different levels of expertise, knowledge, abilities, learning styles, motivation, and competency. Thus, educational recommendations should be personalized to the highest degree.

Most existing RS for the education domain are ratings based, which are generic. While recommending learning materials, the system needs to consider the inherent qualities of the learners to improve personalization. Based on this belief, a prototype was developed for recommending edX MOOCs. The prototype is based on an adapted version of the FSLSM LS model and works exclusively with edX courses. In FSLSM model, students LSs indicate which learning methods (LMs) they prefer. Using this theory, the prototype scans each course to search for learning objects (LOs) which support the preferred LMs of the student and recommend courses with good support. In practice, this is implemented using a scoring mechanism.

For evaluating the prototype, a user study was conducted in an experimental setup and student feedback was used for computing ARHR scores of each recommendation. The ARHR scores were computed for the prototype's recommendations and was compared with scores for ratings based recommendations. It was observed that, the mean ARHR value as per ls_score was obtained as 0.6464, but it doesnt exceed the standard of ratings based recommendations ,which has a mean ARHR score of 0.6896. Based on the results, it can be said that the prototype holds promise, but requires more improvement.

While developing this prototype, valuable lessons were learned, based on which a lot of misconceptions were cleared. In future, it is desired to integrate the prototype to support courses of other MOOCs providers. Also, for learning styles recognition, machine learning based automatic LS recognition is desired to be explored

REFERENCES

[1] G.Pask, "Styles and strategies of learning.", British Journal of Educational Psychology, Vol. 46, pp. 128-148, 1976

[2] P.Honey & A.Mumford, "The manual of learning styles", Maidenhead: Peter Honey, 1982

[3] Kolb, D. A., "Experiential learning: Experience as the source of learning and development." Englewood Cliffs, NJ: Prentice-Hall, 1984.

[4] R.M. Felder and L.K. Silverman. "Learning and teaching styles in engineering education." Engineering education 78.7 (1988): 674-681.

[5] B.A. Solomon and R.M. Felder. "Index of learning styles." Raleigh, NC: North Carolina State University. Available online (1999).

[6] M. Deshpande and G. Karypis, "Item-based top-N recommendation algorithms," ACM Transactions on Information Systems, vol. 22, no. 1, pp. 143177, Jan. 2004.

[7] R.M.Felder & J.Spurlin, "Applications, reliability and validity of the index of learning styles.", International journal of engineering education21.1 (2005): pp. 103-112.

[8] P. Garcia, A. Amandi, S. Schiaffino, and M. Campo, "Evaluating Bayesian networks' precision for detecting students' learning styles," Computers & Education, vol. 49, no. 3, pp. 794 808, 2007.

[9] F. Ricci, L. Rokach, and B. Shapira, "Introduction to Recommender Systems Handbook," Recommender Systems Handbook, pp. 135, May 2010.

[10] A. Ortigosa, P. Paredes, and P. Rodriguez, "AH-questionnaire: An adaptive hierarchical questionnaire for learning styles," Computers & Education, vol. 54, no. 4, pp. 9991005, 2010.

[11] G. Shani and A. Gunawardana, "Evaluating Recommendation Systems," Recommender Systems Handbook, pp. 257297, May 2010.

[12] D. H. Park, H. K. Kim, I. Y. Choi, and J. K. Kim, "A literature review and classification of recommender systems research," Expert Systems with Applications, vol. 39, no. 11, pp. 1005910072, 2012.

[13] F.M.Hollands,, and D. Tirthali. "MOOCs: Expectations and Reality. Full report." Online Submission (2014).

[14] M. M. El-Bishouty, T.-W. Chang, S. Graf, K., and N.-S. Chen, "Smart e-course recommender based on learning styles," Journal of Computers in Education, vol. 1, no. 1, pp. 99111, 2014.

[15] F. Isinkaye, Y. Folajimi, and B. Ojokoh, "Recommendation systems: Principles, methods and evaluation," Egyptian Informatics Journal, vol. 16, no. 3, pp. 261273, 2015.

[16] A. Klanja-Milievi, M. Ivanovi, and A. Nanopoulos, "Recommender systems in e-learning environments: a survey of the state-of-the-art and possible extensions," Artificial Intelligence Review, vol. 44, no. 4, pp. 571604, 2015.

[17] R.Cook, J.Kay, and B.Kummerfeld. "MOOClm: user modelling for MOOCs." International Conference on User Modeling, Adaptation, and Personalization. Springer, Cham, 2015.

[18] F.Bousbahi and H. Chorfi. "MOOC-Rec: a case based recommender system for MOOCs." Procedia-Social and Behavioral Sciences195 (2015): 1813-1822.

[19] G. Paquette, O. Mario, D. Rogozan, and M. Lonard, "Competency-based personalization for massive online learning," Smart Learning Environments, vol. 2, no. 1, 2015.

[20] J. Bernard, T.-W. Chang, E. Popescu, and S. Graf, "Using Artificial Neural Networks to Identify Learning Styles," Lecture Notes in Computer Science Artificial Intelligence in Education, pp. 541544, 2015.

[21] N.Hood and A.Littlejohn. "Quality in MOOCs: Surveying the terrain." (2016).

[22] S. Benhamdi, A. Babouri, and R. Chiky, "Personalized recommender system for e-Learning environment," Education and Information Technologies, vol. 22, no. 4, pp. 14551477, 2016.

[23] S. Wan and Z. Niu, "A learner oriented learning recommendation approach based on mixed concept mapping and immune algorithm," Knowledge-Based Systems, vol. 103, pp. 2840, 2016.

[24] D. Mouromtsev, A. Romanov, D. Volchek, and F. Kozlov, "Metadata Extraction from Open edX Online Courses Using Dynamic Mapping of NoSQL Queries," Proceedings of the 25th International Conference Companion on World Wide Web - WWW 16 Companion, 2016.

[25] "Massive open online course" Wikipedia, 12-Jul-2017. [Online]. Available: [Accessed: 15-Jul-2017].

[26] J. Bernard, T.-W. Chang, E. Popescu, and S. Graf, "Learning style Identifier: Improving the precision of learning style identification through computational intelligence algorithms," Expert Systems with Applications, vol. 75, pp. 94108, 2017.

[27] "Creating a Course with edX Studio", MOOC, Online : https://www.edx.org/course/creating-course-edx-studio-edx-studiox-0

[28] "JSON - Introduction," JSON Introduction. [Online]. Available: $https : //www.w3schools.com/js/js_json_intro.asp$. [Accessed: 15-Jul-2017].

[29] "EdX Research Guide," EdX Research Guide EdX Research Guide documentation. [Online]. Available: http://edx.readthedocs.io/projects/devdata/en/latest/. [Accessed: 15-Jul-2017].

[30] "What is RESTful API? - Definition from WhatIs.com," SearchCloudStorage. [Online]. Available: http://searchcloudstorage.techtarget.com/definition/RESTful-API. [Accessed: 15-Jul-2017].