Expert Systems With Applications 187 (2022) 115926

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Review

A literature review of implemented recommendation techniques used in Massive Open online Courses

Asra Khalid [a,*], Karsten Lundqvist [a], Anne Yates [b]

[a] Victoria University of Wellington, School of Engineering and Computer Science, Wellington, New Zealand
[b] Victoria University of Wellington, School of Education, Wellington, New Zealand

ARTICLE INFO

Keywords:
Massive Open Online Courses (MOOCs)
Recommender systems (RSs)
Recommendation techniques
Evaluation techniques and classification framework

ABSTRACT

Massive Open Online Courses (MOOCs) are receiving attention from learners because MOOCs enable them to satisfy their learning needs through an open, participatory, and distributed way. With the increased interest from learners, the number of MOOCs available is increasing which has increased options for learners. This as a result has created the need for recommendation systems that help learners select suitable MOOCs. This literature review covers analysis of recommender systems (RSs) that have been implemented in MOOCs with the goal of providing insights on the trends reported in the academic literature on recommender systems in MOOCs. The review discusses different recommendation techniques, recommendation types and evaluation techniques that have been used and reported on. This review includes research work over eight years, i.e. from 1st January 2012 to 17th November 2020. After the filtering process, 67 papers were selected from journals and conferences from four academic databases (i.e., IEEE, ACM, Science Direct, and Springer). A framework is designed that classifies literature on the basis of both design and evaluation aspects of RS in MOOCs. This review concludes by highlighting gaps found in the literature.

1. Introduction

Since the advent of the first Massive Open Online Course (MOOC) in 2008 (Downes, 2008), MOOCs have added a new dimension to education systems with the main attraction being access to free open education courses. The number of MOOCs and the number of students registered in MOOCs are growing every year with MOOCs using a number of platforms such as edX, Coursera, Udacity, NetEase and iCourse (Kang, 2014). By the end of 2019, more then 900 universities were offering MOOCs with 13 500 courses available, with around 110 million students registered (Dhawal Shah, 2020b). After the COVID-19 pandemic of 2020 an increase in the online education trend resulted in a surge in the number of new user registrations and the introduction of new courses by different universities. Approximately 25%–30% more registered users joined online courses after the pandemic (Dhawal Shah, 2020a). By the end of 2020, there were more than 950 universities offering MOOCs with 16.3k courses available online and around 180 million new learners registered on them (Dhawal Shah, 2020c). With such a large number of courses available, learners face the problem of selecting courses without being overwhelmed with multiple learning choices (Zhang et al., 2018), which has been termed as an information overload (Raval & Jani, 2016) problem. Information filtering systems, also known as recommender systems(RSs), can overcome this problem by helping learners find suitable courses from the array of available resources.

With the increased usage of MOOCs, data produced by MOOCs is also increasing and this data contain information about the interests and behaviours of learners (Lundqvist & Warburton, 2019). Recommender systems have been used widely by commercial and social platforms (Zhang et al., 2018) and can use educational data to provide recommendations to learners (Ricci et al., 2010). One purpose of these systems is to help the learner by recommending different related learning objects or elements. In addition to improving the learners' experience, these systems have also played a vital role in increasing the popularity of MOOCs.

Recommender systems are intelligent filtering applications that help users to filter out items or information according to their requirements or interests from a large number of services or products. Recommender systems make it easier for users to obtain related information even for items for which they have little knowledge or experience. Some RS provide related information in the form of a priority list, where products or services that are closer to user's interest are placed in higher priority compared with resources which are of lesser interest to the user. For example, the GroupLens system (Resnick et al., 1994) is designed on the assumption that each time a user reads a Usenet

* Correspondence to: Victoria University of Wellington, School of Engineering and Computer Science, Kelburn, Wellington 6012, New Zealand.
E-mail addresses: khalidasra@myvuw.ac.nz (A. Khalid), karsten.lundqvist@vuw.ac.nz (K. Lundqvist), anne.yates@vuw.ac.nz (A. Yates).

https://doi.org/10.1016/j.eswa.2021.115926
Received 7 January 2021; Received in revised form 5 July 2021; Accepted 16 September 2021
Available online 2 October 2021
0957-4174/

News article they will give an opinion. The system will use all opinions as a rating and from these ratings, people with the same rating are considered like-minded, so they can be used to predict the ratings of each other.

RS can be divided into two broad categories (Manouselis et al., 2012) of 'Collaborative Filtering (CF)' and 'Content-based (CBF)' RS. There is a further type called 'Hybrid (HB)' RS that contains characteristics of both collaborative filtering and content-based RS. CF based RSs perform recommendations on the assumption that people who had similar tastes in the past will make the same choice in the future. This is similar to real life scenarios where we have to choose something from multiple available options and we consider recommendations of family and friends who have similar interests (Dakhel & Mahdavi, 2013). CBF RSs consider the profile of items and users to perform recommendations. Profiles include different characteristics of users and items, for example user (age, gender, education, residency area etc.) and item (actor, genre, category, type etc. in the case of a Movie item). These RS analyse the profile of items rated by a user and try to design a model that reflects the interests of the user. This model is used to recommend new items to the user (Lops et al., 2011).

After analysing the literature on RS in MOOCs, we concluded there is a need for a systematic synthesis of literature that summarizes research work and points towards future research options. While reviewing the literature on RS in MOOCs, we found that RSs in MOOCs is an interdisciplinary research field which is of interest to both technical and non technical researchers. After a detailed synthesis we decided to divide this literature into two fields for review, one which summarized the literature for non technical researchers (Author Ref. Khalid et al., 2020), for example researchers from distance learning and education, and the other which will discuss RS in MOOCs from a technical point of view.

To analyse the literature from a technical point of view, we have reviewed the state of the art of RS in MOOCs. The objective of this literature review is to summarize existing work in this field in order to identify gaps and areas in the design and implementation of RS in MOOCs that can help in future work. We reviewed work between January 1st 2012 and November 17, 2020 in the English language only. We chose 2012 as the starting year because it was declared as the "Year of the MOOC" by the New York Times (Pappano, 2012) and from this year onwards the publication of peer-reviewed research on RSs in MOOCs started. We have designed a classification framework which includes both design and evaluation aspects of RS in MOOCs. In addition, we have highlighted the key subject areas of MOOC datasets used in the research. To the best of our knowledge, this is the first literature review in this area that discusses the technical aspects of implemented recommender systems in MOOCs. This literature review not only highlights the trends in the existing literature in this field but also discusses the gaps and possible new research lines.

The remainder of this paper is organized as follows: Section 2 describes the data collection and methodology used to classify the literature work. Section 3 analyses the literature according to the classification framework. Section 4 presents trends found in the literature by discussing each recommendation type found, concluding remarks and research gaps are presented in Section 5.

2. Data collection

Data collection is crucial step in a literature review as it is the basis of the analysis, so we defined a set of rules to set criteria to include or exclude papers. These rules are based upon five significant points: (a) keywords, (b) time period, (c) sources, (d) publication type and (e) work type. 'Keywords' are used to find related published work from specific 'sources', 'time period' refers to the specific duration in which papers were published, 'publication type' refers to the type of research paper (i.e. journal article, conference paper, book chapter, and review article) and 'work type' refers to type of research work in the publication we have considered only those papers in which RS in MOOCs have been implemented and evaluated. The rationale behind the last criteria is to make sure we reviewed only those works in which a RS has been implemented and evaluated not just proposed. This rule allowed us to report trends in evaluation metrics and baselines used for different types of RS in MOOCs. The following section explains these rules in more detail.

- Keywords: This review involved a combination of two concepts; Massive Open Online Courses and Recommender Systems, therefore, we used the following search terms: "Massive Open Online Courses" AND "MOOCs" AND "Recommender System". We added "RS", a common abbreviation for recommender system, along with "Recommender System" to represent Recommender System in the search term however adding "RS" resulted in many unrelated papers. Similarly, we used "MOOC" instead of "MOOCs", but this also resulted in many unrelated papers. This may be because MOOC is also used as an abbreviation for other terms, for example, "Multiple Optical Orthogonal Code Sequences" and "Management of Organizational Change". We also used "Adaptive MOOCs" and "Personalized MOOCs" along with "Recommender System" and "Massive Open Online Courses". With "Personalized MOOCs", we only found one related paper which was already in our database. Whereas, the term "Adaptive MOOCs" resulted in seven papers, but they were already part of our database. Most of the unrelated papers in the latter case were about making MOOCs adaptive and not about recommending any resource or service to the users.
- Time Period: The research period was publications from January 1st 2012 to 17 November 2020. We could not find any published work in 2012 but we kept it our starting period as published work on MOOCs started in this year.
- Sources: To define the sources of research, we followed the same methodology as Liyanagunawardena et al. (2014). The initial searching was in Google Scholar, followed by selected academic databases to find related research work using the same search terms. Four scientific databases were used to search relevant papers: Science Direct, Association for Computing Machinery (ACM) Portal, Institute of Electrical and Electronics Engineer (IEEE) Xplore and Springer Link.
- Publication type: Peer reviewed conference papers, journals and books chapters were included in this literature review.
- Work Type: We have considered only those papers in which RS in MOOCs have been implemented and evaluated using some metrics.

To summarize, we considered only those papers in which a RS for MOOCs had been implemented and the results presented. The first step was to skim the contents of the papers to check whether they were relevant to be selected for deeper review. During the review process, the articles selected were studied in more detail, and further unrelated papers were discarded.

The initial searching on Google Scholar using the above mentioned keywords resulted in over 300 hits, but after initial screening, only 67 papers were found to specifically report on RSs in MOOCs that have been implemented and evaluated, see Table 1 for details. Table 1 shows result of hits and related papers found while searching databases. It can be observed that 21 papers where found while searching in Google Scholar that were not published in any of the selected databases. In the discarded papers, authors had only either proposed the frameworks or discussed the need for RS in MOOCs.

2.1. Data analysis

The yearly distribution of relevant papers is shown in Fig. 1. There has been an increase in number of publications on RS in MOOCs since 2012, with a fall in 2018 and again in 2020. The reason for the fall in

Table 1
Distribution of papers found in academic databases.

Acadamic DB	Hits	Related papers
Springer	177	17
IEEE	52	14
ACM	31	13
SDI	44	2

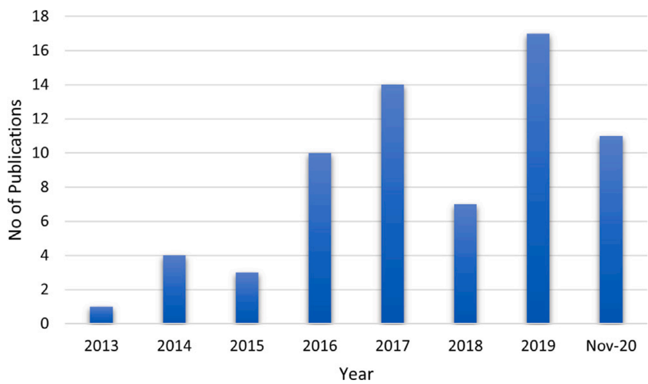

Fig. 1. Distribution of year-wise implemented paper found in literature.

2018 is not obvious, but for 2020 there could be two reasons; first we have not considered the whole year of 2020 and secondly the pandemic situation due to COVID-19 may have affected the research work.

2.2. Classification framework

After analysis of the selected papers, a classification framework was created to better understand the methodologies and trends used in the development of RSs in MOOCs. Based on the analysis of the selected papers, two broad categories were created, i.e Design and Evaluation. The Design category classifies papers on the basis of methods and techniques used in the design of RS and type of recommendations. The Evaluation category classifies papers on the basis of the dataset, baselines and metrics used for the evaluation of the RS.

2.3. Limitations

Limitations of this literature review are that we only considered:

- articles published between January 2012 and 17 November 2020. We cannot eliminate the chance that there may be related conference papers that were presented before 17 November 2020 but were not available online until after 17 November 2020.
- four academic databases and Google Scholar.
- peer reviewed journal articles, conferences, and book sections.
- papers in which the recommender system for MOOCs had been implemented and evaluated.
- only those articles that were published in English. While searching on Google Scholar and performing "reference chaining", we found related articles in other languages such as French. Therefore, there could be relevant non-English articles which we did not include.

The Google Scholar search returned over 600 items. These items included websites, blogs, videos, etc. (5 November 2020). However, we did not analyse these results and did not include these resources because they are subjective and usually not considered for peer review.

3. Analysis using the classification framework

In this section, the distribution based on the proposed classification framework is presented. Fig. 2 shows the classification framework used in this study.

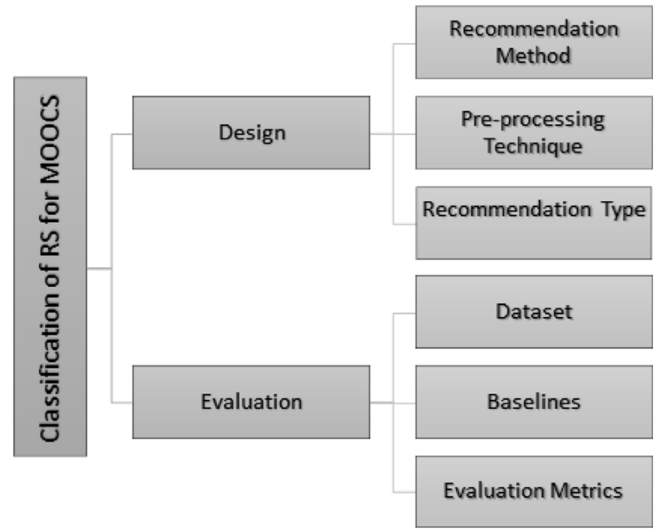

Fig. 2. Classification framework for literature of recommender systems in MOOCs.

3.1. Design

In this section we have classified the literature according to three criteria.

1. Recommendation Type: In this classification, we observed the area of the MOOC for which the RS is implemented. RSs can be used in different areas of MOOCs such as course, peer, learning element, teacher, and thread recommender.
2. Pre-processing Technique: We have classified papers on the basis of pre-processing techniques that were used on the dataset to prepare data for recommendations.
3. Recommendation Method: This classification refers to the type of recommendation method (i.e CF, CBF, HB) used to design the RS.

3.1.1. Recommendation types presented in existing RS in MOOCs

In our systematic literature review (Khalid et al., 2020), we found that literature on RS in MOOCs can be broadly categorized into seven main types on basis of the area of MOOC for which the RS is designed. These types are:

- Course Recommender: Only involves course recommendation.
- Thread Recommender: Thread recommender involves thread/discussion, question recommendation, lecture video clip recommendation in answer to thread questions, and question tag recommendations.
- Peer Recommender: Social interactions are a key factor in successful learning and peer recommender involves systems that are recommending related peers or fellow learners to interact with instead of recommending a learning resource or another class to follow. It uses demographics and progress made in course for recommendations.
- Learning Element Recommender: Learning element recommender includes learning activities, suggestions on study, video lectures, next page recommender, source, personalized question, and learning path recommender.
- MOOC Provider/Teacher Support Recommender: This involves curriculum recommendations, news of MOOCs, and MOOC provider feedback.
- Student Performance Recommender: Student performance recommender involves jobs, grades, student difficulty based, student dropout, work plan, and paid task recommenders.

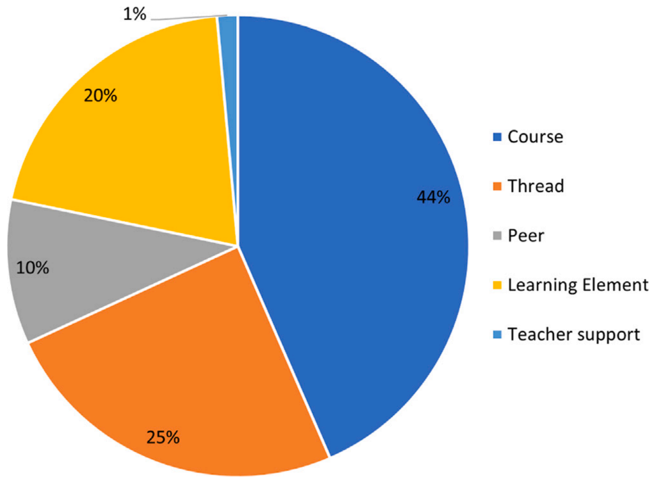

Fig. 3. Distribution of different recommendation types in MOOCs.

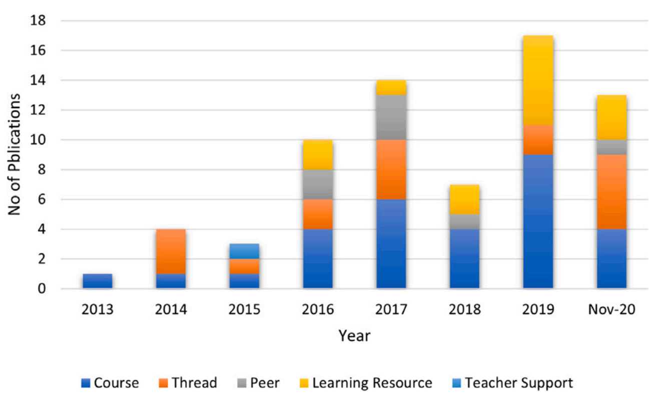

Fig. 4. Yearly distribution of recommendation type in Massive Open Online Courses (MOOCs).

- Others: Others category involves improved and personalized MOOCs, adaptive content, and special user recommender systems.

In the literature on RS that have been implemented and evaluated in MOOCs, we found only five categories as compared to Khalid et al. (2020) (i.e Course Recommender, Thread Recommender, Peer Recommender, Learning Element Recommender and Teacher Support Recommender). Fig. 3 shows the distribution of implemented recommendation types during the considered time frame. It shows that most of the work during that time was focused on course RS.

Fig. 4 shows the yearly distribution of RS types that have been reported in the literature. There was only one paper in 2013 after which a gradual increase in the number of papers reporting RS that have been implemented can be observed with 2017 having the highest number of publications. Throughout the time frame, course recommender systems remained the most popular area of research, but by 2019, Learning Element RS started to gain the interest of researchers. Recommendation types are further discussed in detail in upcoming sections with respect to pre-processing techniques (Section 3.1.2), recommendation methods (Section 3.1.3), baseline (Section 3.2.3) and discussion (Section 4).

3.1.2. Pre-processing techniques used for existing recommender systems in MOOCs

In real world scenarios, generally data is sparse, noisy or inconsistent, for example lacking certain attribute values, containing certain outliers or having certain discrepancies. Data pre-processing is a well established method to resolve such problems. It helps in transforming raw data into an understandable format for the system. It is an important step in recommendation system design because well prepared data improve the recommendation system's efficiency and accuracy.

Researchers have used different techniques for the pre-processing purpose for the implementation of RSs in MOOCs. Fig. 5 shows the distribution of different pre-processing techniques that have been used and reported on. The literature shows a trajectory towards extensive use of Graph based techniques, classification and clustering. We observed an increase in consideration of the behaviour and the context of learners for recommendations. Graph based techniques help in sequencing behavioural and context-based data that can be used for recommendation.

Fig. 5 shows distribution of pre-processing techniques used for different recommendation types.

Graph based Techniques

From Fig. 5 it can be observed that graph based pre-processing techniques are used for course, thread, and learning element recommender systems. From the literature, we observed the use of the Markov decision process, social network, Monte-Carlo Tree and matrix factorization with graph based techniques.

- Course Recommender: For course recommendation (Fazeli et al., 2016; Yang & Jiang, 2019) have maintained user–course graph by creating a directing node from a user to a course that a user has learned while (Agrebi et al., 2019; Zhang, Hao et al., 2019) used the Markov Decision process. Zhang, Hao et al. (2019) used Markov Decision process to remove unrelated data from the dataset and Agrebi et al. (2019) used this process in a layered system to select the best matched courses for the learners and in addition they strived to decrease the number of layers in the system. Hou et al. (2018) used the Monte-Carlo Tree to maintain course clusters of similar courses and also for personalized course recommendations to students on the basis of the history of similar students.
- Thread Recommender: Chen and Epp (2020) used tokenization to remove the stop words and punctuation, after that the posts are maintained as Bag of Words (BoW) and TF-IDF is used to choose the keywords. Subsequently (Chen & Epp, 2020) maintained the graph of thematically similar posts and used them for recommendations. Mi and Faltings (2016a, 2016b, 2017) utilize the Markov model for thread RS to maintain actions of users in the form of context trees where root node is the most generic context which is split into the subsets with every new action from the user.
- Learning Element Recommender: Liu and Li (2019) maintains course–course and learner–learner networks to use them in their learning path based recommendation algorithm. Pardos et al. (2017) maintained learning path as the Markov decision process and used them to recommend next page to learners on the basis of their previous actions.

Matrix Factorization (MF)

Matrix factorization is also used for pre-processing of course, thread recommender, and learning element systems.

- Course Recommeder: Le et al. (2020) combined the Deep Structured Semantic Model (DSSM) (Huang et al., 2013) and matrix factorization for course recommender systems. Panagiotis and Dimitrios (2018) extended the existing MF objective function by adding user skills and course skill matrix along with course and user matrix in the recommendation problem formation. They maintained two matrices, a user–course matrix and a course–user skill matrix and used both of these matrices for the course recommendation process. Zhang et al. (2017) converted user and course attributes into a user feature vector and combined it with the user behaviour feature vector. Zhang et al. (2017) used Deep Belief Networks (DBN) with collaborative filtering for course

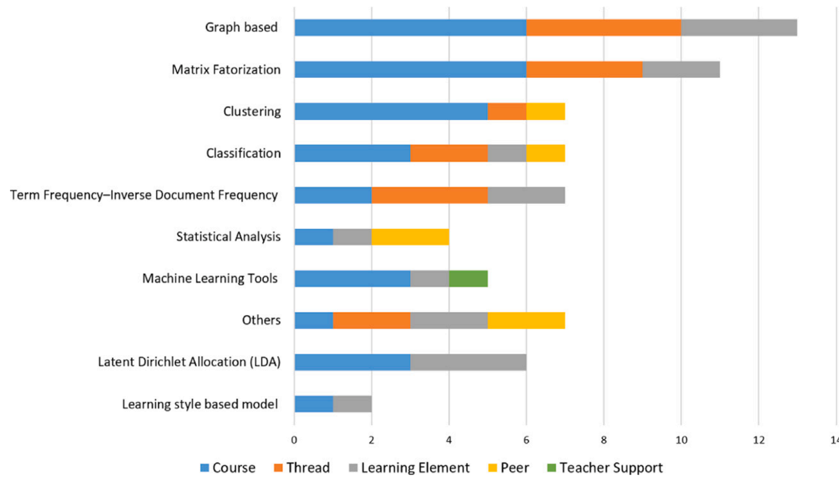

Fig. 5. Distribution of pre-processing technique used for RS in MOOCs with respect to recommendation types.

recommendation. He et al. (2017) maintained a user item rating matrix and used it with collaborative filtering and content based filtering.

- Thread Recommender: Yang, Adamson et al. (2014) designed the context-aware matrix factorization model in which they included student features, question features and student implicit feedback for matrix formation and used them for question relevance calculation in thread RS. Yang, Piergallini et al. (2014) and Yang, Shang et al. (2014) used context and time slice with matrix formation to decrease thread overload by defining a threshold on number of threads recommended to a particular user.
- Learning Element Recommender: For learning element based recommender systems, Sidi and Klein (2020) maintained item and user interaction as a deep learning model instead of inner product of matrix factorization and they introduced a neural network architecture in matrix factorization. Item and user are maintained in the form of vectors which are then connected with embedded layers of neural networks. Gong et al. (2020) modelled the MOOCs platform data as a heterogeneous information network (HIN) where HIN consists of nodes and edges. HIN construct relations between different nodes on the basis of a mapping function e.g if two different students are registered in the same course, there will be a relationship between these two students. Campos et al. (2020) used non-negative matrix factorization with topic modelling techniques to find similar MOOC providers and used them with content-based recommender system for recommendations. Zhang, Huang et al. (2019) used a user–course feature vector matrix as an input to the deep learning system.

In the literature, most researchers have combined matrix factorization with deep learning. We also observed that in the scenarios of MOOCs where more than one attribute of user and course are required, non-negative matrix factorization and an extension of matrix factorization objective function is used.

Clustering

- Course Recommender: Fig. 5 shows that clustering is mostly used for course recommender systems. Wang et al. (2017) tried to overcome the shortcomings of collaborative filtering by designing a multi attribute weight based algorithm and they clustered learners on the basis of different attributes (i.e language, platform etc.). Obeidat et al. (2019) clustered students on the basis of their course choice and then used these clusters for the generation of rules through association rules mining. Fauzan et al. (2020) perform data normalization, cleaning and K-mode clustering and then used these clusters with association rule mining.
- Thread Recommender: Kardan et al. (2017) designed a hybrid approach to utilize both users' posts and users' similarities to form a recommendation problem. They identified similar users on the basis of their interests and posts in threads by using social network analysis and similar posts through association rule mining.
- Peer Recommender: Labarthe et al. (2016) carried out an experimental study to observe the effect of peer recommender systems on students. For this study, they clustered students on the basis of their behaviour and they then analysed the effect of the recommender system on different groups.

Researchers have mostly combined clustering with other techniques, such as association rule mining, content based filtering, data mining techniques etc., because this obtains better results. We observed that clustering has not been used for learning element recommender systems. All of the work on learning element recommender systems has used deep learning, neural network or machine learning for pre-processing and recommendation system design. These systems are not online as they require extensive training. Despite learning element recommender systems needing a large amount of data there is an opportunity to investigate traditional recommender system techniques to design online recommender system for learning elements.

Classification

In the literature, classification is used for all types of recommender systems except teacher support recommender systems.

- Course Recommender: Yanhui et al. (2015) used a Bayesian classifier to classify courses on the basis of course information given in the introduction, syllabus etc. then used this information with collaborative filtering to recommend related courses to users by checking similar user hybrid approach. Jain and Anika (2018) divide learners in two categories i.e. active and passive and then use the KNN classifier for active learners and random forest for passive learners. Hilmy et al. (2019) classified courses on the basis of their video lecture styles which they used with the CNN model.
- Thread Recommender: Agrawal et al. (2015) classify posts in threads on the basis of sentiment, urgency and other variables and then they used information retrieval to map questions in post to a related video clip from the lecture. Babinec and Srba (2017) used supervised multi-label classification on the basis of the text of questions in a thread to form a recommendation problem.
- Peer Recommender: Bouchet, Labarthe, Bachelet et al. (2017) observed the behaviour of users who used peer recommender system to understand reason of their usage. They divided users into three classes based on their behaviour data.

- Learning Element Recommender: Kopeinik et al. (2016) carried out a comparative study which included existing algorithms to ascertain which algorithm worked best in which learning environment.

TF-IDF

TF-IDF has been used when researchers have considered using course introduction, thread posts, text or video content for recommendations. TF-IDF is generally used with Bag of Words (BOW), tokenization and similarity measurement techniques.

- Course Recommender: Piao and Breslin (2016) used TF-IDF to identify skills of learners from their Linkedin profiles and course information, and used this information for recommendations.
- Thread Recommender: Babinec and Srba (2017) utilized TF-IDF for keyword extraction from thread posts and then find tags for each post and recommend tags of posts to learners. Macina et al. (2017) used tokenization, BOW, and TF-IDF to create question profiles. They also maintained user profile and match question profile with user profile to recommend related questions for users to answer. Rahma and Koutheair (2019) utilized TF-IDF to find keywords in questions and then used cosine similarity to find similar posts and video snippets to learner for his question. Trirat et al. (2020) used TF-IDF with classification and deep learning they used Natural Language Processing (NLP) as well. They recommend snippets of lecture videos to users in response to their question in a thread.
- Learning Element: Hajri et al. (2018) used TF-IDF to create learner profiles by using course information.

Statistical Analysis

Statistical analysis involves the calculation of ratings for items based on user behaviour and reciprocal score calculation for peer recommender systems. It also involves pre-processing of data for matrix factorization.

- Course Recommender: Garg and Tiwari (2016) calculated implicit rating by dividing the total time spent on a course by the total expected time required to complete the course, then used K-Means for clustering.
- Peer Recommender: Prabhakar et al. (2017) calculated similarity between two users by considering their common interests, age, gender and location. They calculated similarity between two users by calculating the distance between each attribute of the users and then taking the mean of all of distances. Potts et al. (2018) calculated reciprocal score for a user on the basis of their interest in a topic, availability in a given time for a study session, interest in the topic and willingness to play a role in the session.
- Learning Element Recommender: Zhang, Huang et al. (2019) pre-processed the user feature, course feature and user activity data by cleaning the data and normalizing it. They then formed a feature matrix to use as an input to the deep learning model

Machine Learning Tools

In the literature reviewed, different machine learning tools are used for pre-processing of datasets.

- Course Recommender: Sakboonyarat and Tantatsanawong (2019) normalized, cleaned the data to modify the missing data and used it for deep learning neural network. Ahera and Lobo (2013) utilized Weka to pre-process data and used it for association rule mining. Zhang et al. (2018) used hadoop to pre-process data and used it for ARM in Spark. Dahdouh et al. (2019) utilized different tools like spark, hadoop and proposed an FP growth algorithm and compared it with existing algorithms and library in spark.
- Learning Element Recommender: Liu and Li (2020) used machine learning tools to crawl and clean learning data from the MOOC platform. After pre-processing they transformed data into relational data structure.
- Teacher Support Recommender: Zhou et al. (2015) proposed the use of natural language processing for improving results of a teacher support recommender system.

Latent Dirichlet Allocation (LDA)

LDA is used for topic extraction from posts, course content and videos of lectures. LDA provides insights into the type of material that helps in understanding the interests of users.

- Course Recommender: Jing and Tang (2017) carried out topic modelling by extracting all the web pages that users visit in the course and finding keywords from those pages and used these topic to find similar users for recommendation. Apaza et al. (2014) used LDA to extract topics from university courses and MOOCs to then recommended similar courses to users.
- Thread Recommender: Lan et al. (2019) used topic modelling along with timescale modelling for decay in post excitement and user interest modelling to create one model and used it for recommendation.
- Learning Element Recommender: Bhatt et al. (2018) and Zhao et al. (2018) used LDA to extract topics from videos and found inter-topic relationships in different videos to recommend related videos to user. Zhang, Zhu et al. (2019) used LDA to extract topic from course content then used Page Rank and weighting method to provide related material to users based upon their history of usage.

Learning style (LS)

Learning style based modelling is also used by some researchers to recommend related course content to user. Researchers have used the Felder–Silverman learning styles model in conjunction with other attributes of the users and courses for making recommendations.

- Course Recommender: Gope and Jain (2017) used the FSLSM model to identify learning styles of users to recommend related courses based on the learning style and requirement of the user.
- Learning Element Recommender: Hajri et al. (2018) used FSLSM and learner knowledge and language to create learner profiles and course attributes to create a course profile for learning resource recommendation.

Others

In this category, are the less common pre-processing techniques and comparative studies of existing work.

- Course Recommender: Boratto et al. (2019) is a comparative study that observed the effect of algorithm bias which used ratings datasets to carry out experiments.
- Thread Recommender: Zhang et al. (2020) filtered out threads with length 1 and that are viewed less than two times from datasets before using it for recommendation. Malgonde et al. (2020) did not describe the pre-processing technique they used in their work.
- Peer Recommender: Jo et al. (2016) filtered course related tweets by using hashtags and then removed unrelated tweets. They categorized users on the basis of their goal statement in different categories. Malgonde et al. (2020) has not clearly mentioned the techniques that they used for pre-processing.
- Learning Element Recommender: Subramanian et al. (2019) has not clearly mentioned any pre-processing technique.

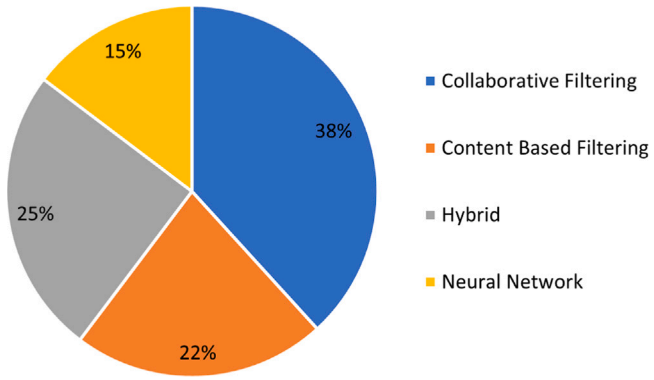

Fig. 6. Distribution of recommendation methods used for RS in MOOCs.

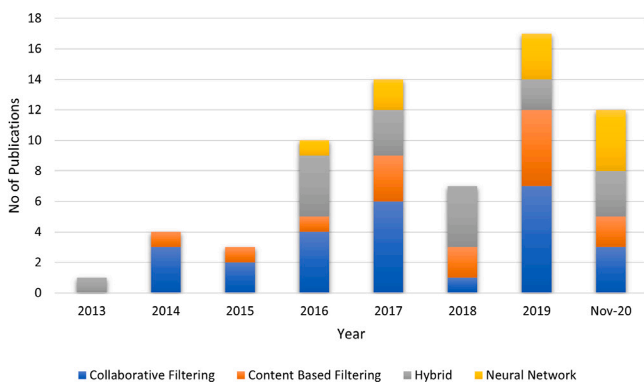

Fig. 7. Yearly distribution of recommendation methods used for RS in MOOCs.

3.1.3. Recommendation methods reported in the literature

Authors have reported different recommendation methods and this section discusses the most common recommendation methods used by researchers.

CF, HB and CBF are the most common recommendation methods reported (Khalid et al., 2020). HB contains research work done by combining two existing recommendation algorithms (Li et al., 2018). Fig. 6 shows the distribution of recommendation methods used for RS in MOOCs. It shows that CF and HB seem to be the most popular among the remaining recommendation methods. In the last few years, researchers have also started exploring neural networks and deep learning algorithms for designing RS in MOOCs. We have described such work as Neural network (NN) .

Fig. 7 shows yearly distribution of recommendation methods used for RS in MOOCs. Throughout the period, researchers tended to rely on CF based recommendation techniques, but from 2016, a trend of using neural networks emerged. These include the use of deep belief networks and Recurrent neural networks.

Fig. 8 shows the trend of usage of different recommendation methods with respect to the type of recommendation. In all types of recommendations, CF is the most common type except for learning element recommender systems. For learning elements more papers report using CBF, HB and NN than those using CF. It is also observed that NN is used mostly for Course, Thread and Learning element recommendation types. One possible reason for this trend is that NN can improve the results in the above mentioned recommendation types by including behavioural attributes of users.

Table 2 shows recommendation methods used, categorized with respect to recommendation type. In the literature most of the work is on course RS.

Course Recommender:

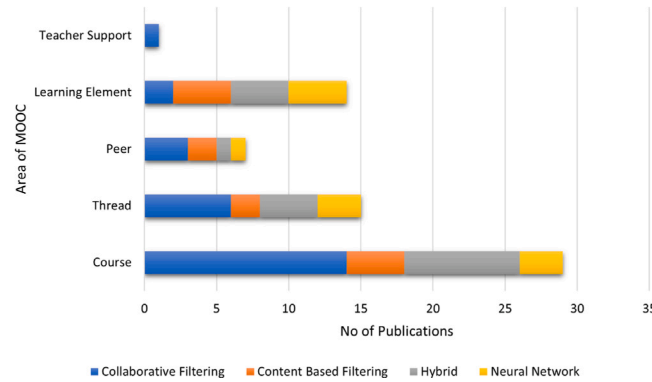

Fig. 8. Recommendation methods used in RS along with recommendation type for MOOCs.

CF is mostly used for the design of course RS. Researchers have used different attributes of learners and courses to improve the results of CF based RS. Zhang et al. (2018) has improved basic Apriori association rule mining to define the rules for every course using course enrolment data and used these rules for recommendation. Whereas (Dahdouh et al., 2019) has used parallel FP growth algorithm for the implementation of association rule mining. They have used spark and hadoop for the implementation of their distributed system. Gope and Jain (2017) utilized Felder and Silverman learning styles to calculate a learning style score of all the courses and matched it with required learner's style for course recommendation. Wang et al. (2017) designed a CF based recommendation algorithm that maintains a user attribute matrix instead of a user item matrix to overcome the CF cold start problem. Jing and Tang (2017) has also used CF but they extracted course information by topic modelling and used that information while building user profile. Pang et al. (2018) designed a CF based recommender system that decreased the drop out rate from MOOCs by recommending courses to the user by considering learning score, learning time and learning intensity as key features to define learners satisfaction. They have used learners' neighbours features and learning series. Learner neighbours' features show the difference between learning items and learning series' behaviours reflect the evolution of learning features about the target learner. Zhang, Hao et al. (2019) designed a basic collaborative filtering RS with a profile revision model. This model allows the system to revise the profile of a user to remove noisy and unrelated courses from a user profile to improve the results of the RS. They used the Markov Decision Process to identify noisy courses. Pang et al. (2019) designed a CF recommendation method based on prerequisites because to complete any course, a user should have adequate knowledge. They designed an RS that recommends courses by considering prerequisite requirements to complete a course. Their RS covers both prerequisite and subsequent learning goal. Garg and Tiwari (2016) used implicit rating derived from user activities in MOOCs and used it for recommendation by using CF.

Apaza et al. (2014) designed a MOOC recommender for college students. They extracted topics from the course content of college courses and MOOCs by using LDA and then compared the topics of MOOCs with the college course and recommended related MOOC courses by using CBF. Piao and Breslin (2016) performed a comparative study to observe the effects of different user profiles on course recommendation. They used content based RS for their studies. They found user profile based on user skills, gave a better result compared with job and education based profile. Yanhui et al. (2015) have utilized both CBF and CF for their RS. They used course information and learner history to classify learner and courses in different groups and then they used traditional CF to merge the course and learner group. Ahera and Lobo (2013) combined simple K-means and association rule mining and compared the results with open source data mining Weka tool. They have discussed other possible options to combine clustering techniques with association rule mining in their work. Hou et al. (2016) designed a big data supporting hierarchical tree based context aware course RS that can work with massive datasets. The algorithm can gradually recommend the most preferred and matched courses to students. They have used distributed-connected storage nodes to work on massive MOOC data.

Thread Recommender:

Most of the work on thread recommender has used a hybrid approach. Yang, Adamson et al. (2014), Yang, Piergallini et al. (2014) and Yang, Shang et al. (2014) designed context-aware matrix factorization by using contextual information of the learner to estimate learner interest in threads in a discussion forum. Mi and Faltings (2016a, 2016b, 2017) designed a context tree by using the Markov chain model to recommend the most relevant thread to the learners. This is the only online recommender system that can recommend threads to users by considering their current context. Kardan et al. (2017) has used social network and association rule mining to design their RS. Babinec and Srba (2017) recommend tags to the learners for their post by using top ‘n’ recommendation methods. They used TF-IDF to divide the text of question in keywords and used classification to divide these tags in groups. This system has some limitations, such as if the course content is updated, the RS requires retraining. Rahma and Koutheair (2019) used CBF to answer user questions by finding related existing answers in threads or recommending clips from lecture videos. They have used supervised classification and data mining whereas, Trirat et al. (2020) used deep neural network, confusion classifier and CBF to built a RS that recommends a set of ranked video snippets for providing answers to the student’s question.

Peer:

Labarthe et al. (2016) is an experimental study in which the authors observed the affects of peer recommender on student performance. They used CF to cluster similar users on the basis of their activities. Prabhakar et al. (2017) and Potts et al. (2018) both used reciprocal score based peer RS. Prabhakar et al. (2017) used age, gender, education and interests to calculate reciprocal score for each user and find similar users on the basis of this score. Potts et al. (2018) designed an algorithm that recommends peer learning sessions based on the preferences and needs of individuals that are interested in providing learning support, seeking learning support or finding study partners.

Learning Element:

In the work on designing learning element RS, authors have mostly used neural networks, data mining and machine learning. Pardos et al. (2017) used recurrent neural networks to recommend next page to the user on the basis of the history of their click streams and time spent on a page; whereas Gong et al. (2020) designed a graph neural network based approach by using both context and content information. They proposed to use extended matrix factorization for tuning the parameters of their algorithm. Sidi and Klein (2020) designed neural collaborative filtering (NCF) model to sequence questions by difficulty in MOOCs to recommend personalized questions for each learner. Zhang, Zhu et al. (2019) designed a personalized MOOC guidance system using weight based algorithm that takes into account course content, student knowledge level and provides a personalized course path to users. They have used cosine similarity to find related course material for learners. They also designed a dropout prediction system to understand the link between student behaviour and dropout. Hajri et al. (2018) designed a CBF based two step learning element RS that recommends resources to learners, once when a user starts a course on the basis of prerequisites for the course, and then at the end of the course on the basis of learner performance in a quiz.

Teacher Support:

Zhou et al. (2015) designed a user friendly mobile application in which students check-in weekly to access their own performance data and they also give ideas to improve the course content. In addition, students can rate other students ideas which can help teachers to improve their course. Zhou et al. (2015) have used Natural language processing in their work to process student ideas.

Tables 3 and 4 shows categorization of evaluation metrics with recommendation method used in the literature. A detailed discussion on these tables is given in Section 3.2.2.

Fig. 9 shows pre-processing techniques used for different recommendation methods. For the pre-processing of data to apply CF, mostly graph based, clustering and matrix factorization is used, whereas for CBF, TF-IDF, LDA and classification is used. For HB every type of pre-processing is used because HB is a combination of two algorithms and the need for a pre-processing technique varies with the type of recommendation method used. For data preparation to apply NN, mostly graph based machine learning and matrix factorization is used. From Fig. 9, it can be observed that graph based, Matrix factorization and machine learning tools are used for all recommendation methods.

3.2. Evaluation

In this section we classify the work on the basis of evaluation methods. We have further divided this into sub categories which are described as follows:

1. Dataset: Through this classification we have highlighted datasets used by researchers for the evaluation of their research work.
2. Evaluation Metrics: We have classified the literature on the basis of evaluation metrics used for the evaluation of the implemented RS.
3. Baseline: In this category, we explored most the commonly used baselines for bench marking.

3.2.1. Datasets used for the evaluation of RS in MOOCs

In previous research datasets from Coursera and edX have mostly been used to evaluate the RSs. In some cases, researchers have used datasets from their own institutions and some have created their own datasets by creating a Small Private Online Course (SPOC).

Fig. 10 shows the different datasets used by researchers. As can be seen, datasets from Edx and Coursera were used in 50% of the literature reviewed. The ‘others’ category includes the less common datasets used for evaluation of recommender systems, these include COCO (Boratto et al., 2019), Merlot (Panagiotis & Dimitrios, 2018), Mace (Panagiotis & Dimitrios, 2018), iCourse (Hou et al., 2018), Open University database openU (Fazeli et al., 2016),QuCryptox (Macina et al., 2017), IBM provides the dataset for Apriori algorithm (Zhang et al., 2018), Ripple platform dataset (Potts et al., 2018), Stanford MOOC Posts Dataset (Agrawal et al., 2015; Rahma & Koutheair, 2019) and KDD (Kopeinik et al., 2016). While analysing the datasets used, we found that there is only one publicly available data set known as the COCO dataset (Dessì et al., 2018) which contains ratings for courses as well as other features of courses.

- **Subject Area of datasets used:** In addition, we analysed the subject areas of these datasets. In most of the works (Agrebi et al., 2019; Apaza et al., 2014; Boratto et al., 2019; Bouchet, Labarthe, Bachelet et al., 2017; Fazeli et al., 2016; Garg & Tiwari, 2016; Gope & Jain, 2017; Hou et al., 2018; Jain & Anika, 2018; Jiang & Pardos, 2019; Kopeinik et al., 2016; Labarthe et al., 2016; Liu & Li, 2019; Mi & Faltings, 2016a, 2016b; Obeidat et al., 2019; Panagiotis & Dimitrios, 2018; Pang et al., 2018; Pardos et al., 2017; Piao & Breslin, 2016; Piedra et al., 2014; Potts et al., 2018; Prabhakar et al., 2017; Wang et al., 2017; Zhang, Huang et al., 2019; Zhang et al., 2018; Zhou et al., 2015) the subject area of the dataset is not noted. Fig. 11 presents the subject areas of MOOCs that have been used for the evaluation of RS in MOOCs. The most commonly investigated subject area found is ’Computer Science’.

Table 2
Recommendation methods used in RS along with recommendation type for MOOCs.

	CF	CBF	HB	NN
Course	Boratto et al. (2019), Dahdouh et al. (2019), Fauzan et al. (2020), Garg and Tiwari (2016), Gope and Jain (2017), Jing and Tang (2017), Le et al. (2020), Obeidat et al. (2019), Pang et al. (2018, 2019), Wang et al. (2017), Yang and Jiang (2019), Zhang, Hao et al. (2019) and Zhang et al. (2018)	Agrebi et al. (2019), Apaza et al. (2014), Hou et al. (2018) and Piao and Breslin (2016)	Ahera and Lobo (2013), Fazeli et al. (2016), He et al. (2017), Hilmy et al. (2019), Jain and Anika (2018), Panagiotis and Dimitrios (2018), Yao et al. (2020) and Yanhui et al. (2015)	Hou et al. (2016), Sakboonyarat and Tantatsanawong (2019) and Zhang et al. (2017)
Thread	Yang, Piergallini et al. (2014), Yang, Shang et al. (2014) and Yang, Adamson et al. (2014)	Agrawal et al. (2015), Babinec and Srba (2017), Lan et al. (2019), Rahma and Koutheair (2019) and Trirat et al. (2020)	Chen and Epp (2020), Kardan et al. (2017), Macina et al. (2017) and Mi and Faltings (2016a, 2016b, 2017)	Zhang et al. (2020) and Trirat et al. (2020)
Peer	Jo et al. (2016), Labarthe et al. (2016) and Prabhakar et al. (2017)	Bouchet, Labarthe, Yacef et al. (2017) and Bouchet, Labarthe, Bachelet et al. (2017)	Potts et al. (2018)	
Learning element	Liu and Li (2020) and Gong et al. (2020)	Bhatt et al. (2018), Campos et al. (2020), Hajri et al. (2018) and Zhang, Zhu et al. (2019)	Elbadrawy et al. (2016), Kopeinik et al. (2016), Subramanian et al. (2019) and Zhao et al. (2018)	Jiang and Pardos (2019), Liu and Li (2020), Pardos et al. (2017) and Zhang, Huang et al. (2019)
Teacher	Zhou et al. (2015)			

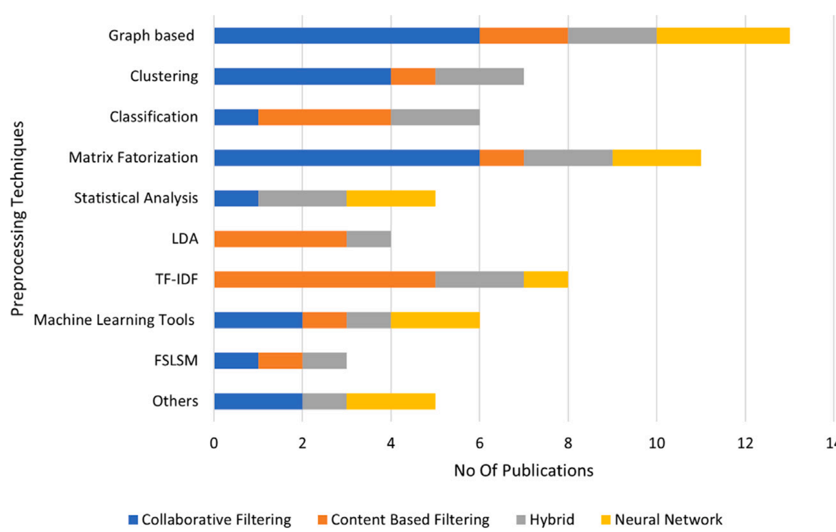

Fig. 9. Recommendation methods used in RS in conjunction with data pre-processing techniques.

3.2.2. Evaluation metrics

To analyse evaluation techniques used in the literature, we classified RSs in MOOCs with respect to recommendation method and metrics used to evaluate its results. We categorized recommendation methods as Collaborative filtering (CF), Content Based filtering (CBF), Hybrid (HB) and Neural Network (NN). We found different types of metrics were used for the evaluation of recommender systems.

Table 3 shows a detailed list of metrics used for the evaluation of CF and CBF based recommender systems. MAP, Recall, Precision, F1, and nDCG is mostly used for the evaluation of these recommendation systems. It can be observed from Table 3 that only one work has discussed the time complexity of their work.

Table 4 shows detailed list of metrics used for the evaluation of HB and NN based recommender systems. Accuracy, Recall, Precision, and nDCG are mostly used for the evaluation of HB and NN based recommendation systems. It is observed that most of the evaluation metrics are common for all the four recommendation methods except MAE and ARHR. MAE is used for evaluation of HB and NN only, whereas, ARHR is used for the evaluation of CF and CBF.

Other than the above mentioned metrics, there are few other metrics which are specific to the MOOCs in RS which are categorized as 'Course Interaction' in Tables 3 and 4. Course interaction metrics measure the interaction of learners with the courses. They include student attendance, final score, student activities, number of completed quizzes and assignments, and the number of students who completed courses. The metrics also include characteristics of the course, such as, topic redundancy, course diversity, matching topics, novelty within section, and coverage of topic.

Other metrics that authors have defined on the basis of their research goals, are categorized as 'Others' in Tables 3 and 4. Hajri et al. (2018, 2018), Hilmy et al. (2019), Yanhui et al. (2015), Zhou et al. (2015) and Subramanian et al. (2019) surveyed the users of RSs, other researchers defined their own objective functions (Jo et al., 2016; Mi & Faltings, 2016b; Yang, Shang et al., 2014; Zhang et al., 2018). To evaluate association rule mining based recommender systems, support, confidence, and recommendation efficiency have been used Fauzan et al. (2020) and Zhang et al. (2018). Mi and Faltings (2017) performed comparison of adaptability of their algorithm with baselines. Zhang,

Table 3
Table showing metrics used in literature to evaluate CF and CBF based recommender systems. (Rows as metrics; and columns as recommendation method).

	CF	CBF
MAP	Dahdouh et al. (2019), Jo et al. (2016), Yang, Piergallini et al. (2014) and Yang, Adamson et al. (2014)	Lan et al. (2019) and Zhang, Zhu et al. (2019)
Time	Yang, Shang et al. (2014) and Garg and Tiwari (2016)	
Accuracy		Apaza et al. (2014), Trirat et al. (2020)
RMSE	Garg and Tiwari (2016) and Boratto et al. (2019)	
MRR	Jing and Tang (2017)	Piao and Breslin (2016)
S@N		Piao and Breslin (2016)
Regret		Hou et al. (2018)
Course interaction		Bouchet, Labarthe, Yacef et al. (2017) and Bhatt et al. (2018)
Recall	Boratto et al. (2019), Dahdouh et al. (2019), Liu and Li (2019), Pang et al. (2018), Prabhakar et al. (2017), Wang et al. (2017), Zhang, Hao et al. (2019) and Pang et al. (2019)	Agrebi et al. (2019) and Babinec and Srba (2017)
Precision	Boratto et al. (2019), Liu and Li (2019), Pang et al. (2018, 2019), Prabhakar et al. (2017) and Yang and Jiang (2019)	Agrawal et al. (2015), Agrebi et al. (2019) and Babinec and Srba (2017)
F1	Pang et al. (2018) and Pang et al. (2019)	Pang et al. (2019), Rahma and Koutheair (2019), Zhang, Zhu et al. (2019) and Trirat et al. (2020)
nDCG	Boratto et al. (2019), Gong et al. (2020), Le et al. (2020), Prabhakar et al. (2017) and Zhang, Hao et al. (2019)	Agrawal et al. (2015), Trirat et al. (2020) and Zhang, Zhu et al. (2019)
ARHR	Gope and Jain (2017)	
Coverage	Obeidat et al. (2019)	Bhatt et al. (2018)
AUC	Boratto et al. (2019)	
Scalability	Zhang et al. (2018) and Zhang, Hao et al. (2019)	Hou et al. (2018)
Others	Fauzan et al. (2020), Jo et al. (2016), Labarthe et al. (2016), Yang, Shang et al. (2014), Zhou et al. (2015) and Zhang et al. (2018)	Bouchet, Labarthe, Bachelet et al. (2017), Campos et al. (2020), Hajri et al. (2018) and Zhang, Zhu et al. (2019)

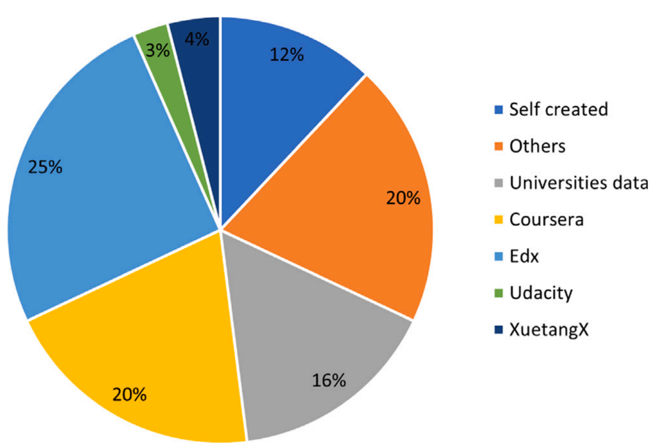

Fig. 10. Datasets used for evaluation of RS in MOOCs.

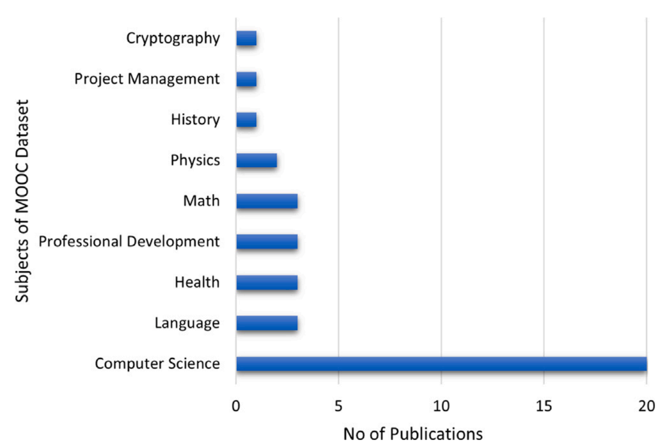

Fig. 11. Subject areas of datasets used for evaluation of RS in MOOCs.

Zhu et al. (2019) used Kendall's coefficient (widely used for ranking models) to evaluate results of their algorithm while (Potts et al., 2018) measured reciprocity of their reciprocal peer recommender algorithm.

3.2.3. Baselines used for evaluation

When evaluating RS in MOOCs, most of the work is bench marked against a baseline, but some researchers (Agrawal et al., 2015; Agrebi et al., 2019; Fauzan et al., 2020; Hajri et al., 2018; Jiang & Pardos, 2019; Jo et al., 2016; Kardan et al., 2017; Kopeinik et al., 2016; Liu & Li, 2019; Malgonde et al., 2020; Rahma & Koutheair, 2019; Sakboonyarat & Tantatsanawong, 2019; Subramanian et al., 2019; Trirat et al., 2020) have shown results of their techniques by using different metrics without comparing those results with any baseline. Other researchers carried out comparative studies using existing techniques to observe the effect of different parameters on the results of their RS (Fazeli et al., 2016; Jain & Anika, 2018).

Table 5 shows different baselines that have been used for bench marking with the recommendation type. For course recommender systems, CF based recommender systems are mostly used as the baseline. To bench mark thread recommender systems, MF based recommender systems are used. For peer recommender systems, case studies and surveys are mostly used for bench marking. Learning element recommender systems are compared with random algorithms and some researchers have used student surveys.

In Table 5, popular recommender systems recommend the most popular items from the dataset independent of the user's interests (Yang, Adamson et al., 2014). Random recommender recommends random items to the user (Bouchet, Labarthe, Yacef et al., 2017). Matrix Factorization (MF) is the class of collaborative filtering algorithm that

Table 4
Table showing metrics used in literature to evaluate HB and NN based recommender systems. (Rows as metrics; and columns as recommendation method).

	HB	Neural network
MAP	Mi and Faltings (2016a) and Macina et al. (2017), Mi and Faltings (2017)	Jiang and Pardos (2019)
Time	Garg and Tiwari (2016)	
Accuracy	Elbadrawy et al. (2016), He et al. (2017) and Jain and Anika (2018)	Hou et al. (2016), Pardos et al. (2017) and Trirat et al. (2020)
MAE	Elbadrawy et al. (2016)	
RMSE	Garg and Tiwari (2016)	Zhang, Huang et al. (2019) and Zhang et al. (2017)
MRR	Macina et al. (2017)	
S@N	Macina et al. (2017)	
Regret		Hou et al. (2016)
Course interaction	Zhao et al. (2018)	
Recall	Fazeli et al. (2016), He et al. (2017), Jain and Anika (2018), Kardan et al. (2017), Kopeinik et al. (2016) and Panagiotis and Dimitrios (2018)	Liu and Li (2020) and Zhang et al. (2020)
Precision	Chen and Epp (2020), Fazeli et al. (2016), Jain and Anika (2018), Kardan et al. (2017), Kopeinik et al. (2016), Panagiotis and Dimitrios (2018) and Sidi and Klein (2020)	Liu and Li (2020) and Sakboonyarat and Tantatsanawong (2019)
F1	Jain and Anika (2018), Kopeinik et al. (2016) and Yanhui et al. (2015)	Trirat et al. (2020)
nDCG	Kopeinik et al. (2016), Macina et al. (2017) and Yanhui et al. (2015)	Trirat et al. (2020) and Zhang et al. (2020)
Coverage	Potts et al. (2018)	
AUC	Jain and Anika (2018) and Potts et al. (2018)	
Scalability	Ahera and Lobo (2013), Mi and Faltings (2016a, 2017) and Potts et al. (2018)	
Others	Hilmy et al. (2019), Mi and Faltings (2016b, 2017), Potts et al. (2018), Subramanian et al. (2019), Yao et al. (2020) and Yanhui et al. (2015)	Malgonde et al. (2020)

converts users' item interaction into two lower dimensional rectangular matrices (Panagiotis & Dimitrios, 2018). K-nearest neighbours (KNN) consider top 'k' nearest items to perform recommendations (Zhang, Huang et al., 2019). Fresh recommender systems consider the most recent items for recommendations. This baseline is normally used for thread recommender to compare results with a recommender system which is recommending the most recent thread to the user (Mi & Faltings, 2016a). Item and user based CF and CBF are also used as baselines. Space Vector Decomposition (SVD) is also a matrix factorization technique which is used to reduce the dimensionality of the dataset (Yang, Shang et al., 2014).

Others baselines include Non personalized baselines (Boratto et al., 2019), common recommendation techniques (cold start and non cold start comparison Elbadrawy et al., 2016, mean of mean Elbadrawy et al., 2016, Top N recommendations Prabhakar et al., 2017, n-gram Pardos et al., 2017, High confidence tree Hou et al., 2016, 2018, item or user average Boratto et al., 2019, Hierarchical Optimistic Optimization tree Hou et al., 2018, Adaptive Clustering Recommendation Hou et al., 2016, 2018 and Directly Content Match (DCM) Yang, Piergallini et al., 2014), Apriori algorithm (Ahera & Lobo, 2013; Obeidat et al., 2019), association rule mining based on Hadoop (Zhang et al., 2018), LDA (Campos et al., 2020), Bayesian personalized ranking (Gong et al., 2020), Factorization Machines (Gong et al., 2020) and researchers previous work i.e variants of their own algorithms (Gong et al., 2020; Macina et al., 2017; Yang, Shang et al., 2014) . Ahera and Lobo (2013) and Dahdouh et al. (2019) carried out a comparative study of their algorithm with Weka and RStudio software. For thread recommender systems researchers have compared their algorithm results with most recent thread recommender, social interaction based thread recommender that recommend threads to learners on basis of their previous social interaction on the forum (Bouchet, Labarthe, Yacef et al., 2017; Lan et al., 2019). For Peer recommender designed using reciprocal score (Potts et al., 2018) used baseline non-reciprocal score to evaluate the results of their systems. To bench mark teacher recommender, Zhou et al. (2015) manually assigned scores to material and compared these with recommended topics and in addition conducted surveys with teachers to compare results of their RS.

Case Studies (C.St) include baselines created by conducting surveys of users (Gope & Jain, 2017; Hilmy et al., 2019), comparing results of the user with and without RSs (Bouchet, Labarthe, Bachelet et al., 2017; Labarthe et al., 2016), post study questionnaires (Bhatt et al., 2018; Zhao et al., 2018) and comparing results of different datasets (Apaza et al., 2014).

4. Discussion

To the best of our knowledge, this is the first systematic literature review based on a classification framework of RSs that have been implemented in MOOCs from January 1st 2012 to 17 November 2020. We classified RSs that have been implemented based on the design of the RS and the evaluation techniques used.

We found five areas of MOOCs for which RS have been developed, implemented and evaluated. In this section we discuss the types and trends of research carried out within each of these areas along with the common attributes of MOOCs and learners that are used for computation of the recommendation. We also present what we consider are gaps in the current research agenda and are therefore areas for future research.

4.1. Course recommender

Most of the work on RSs in MOOCs has focused on the design and implementation of course RSs (Khalid et al., 2020). Reasons for this trend could be the availability of large data for the evaluation of these RSs and that these systems can help MOOC providers in increasing their enrolments.

While designing course RSs, different attributes of learners are considered. We observed that when using historical data of learners researchers have preferred association rule mining (Ahera & Lobo, 2013; Fauzan et al., 2020; Zhang et al., 2018, 2017). Jain and Anika (2018) divided learners in two categories based on their involvement rate in the MOOC - one active and the other passive and they used machine learning for active learners and CF for passive learners. Pang et al. (2018) designed a course recommendation algorithm based on user interest, demographic profiles and course prerequisite relation. Gope and Jain (2017) considered learning styles of learners while recommending courses for which they used the Felder–Silverman Learning Style Model (FSLSM) to create profiles of learners. Hilmy et al. (2019) used learning style and sentiment analysis of users' comments on threads to recommend related courses to learners.

Fazeli et al. (2016) designed a RS for open online courses instead of a closed course RS because in open online courses there is a lack of user profile and historical data. For recommendations, they utilized the user's online activities and interaction data and their RS produced satisfactory results. Zhang et al. (2017) used the Deep Belief network for feature extraction and prediction classification. Their system mines learners interests through users features and course features and uses them for recommendations.

Table 5
Baselines used for evaluating the recommender system and the area of the MOOC where these recommender systems are used.

Publication	Popular	Random	MF	KNN	Others	Fresh	CBF	CF	SVD	C.St
Course recommender										
Zhou et al. (2015)					✓					
Piao and Breslin (2016)	✓									
Hou et al. (2016)					✓					
Gope and Jain (2017)										✓
Wang et al. (2017)								✓		
He et al. (2017)								✓		
Jing and Tang (2017)		✓						✓		
Hou et al. (2018)					✓					
Zhang et al. (2018)					✓			✓		
Pang et al. (2018)								✓		
Boratto et al. (2019)	✓	✓			✓					
Ahera and Lobo (2013)					✓					
Panagiotis and Dimitrios (2018)			✓						✓	
Yanhui et al. (2015)							✓			
Garg and Tiwari (2016)								✓		
Apaza et al. (2014)										✓
Dahdouh et al. (2019)					✓					
Obeidat et al. (2019)					✓					
Zhang et al. (2017)				✓			✓		✓	
Zhang, Hao et al. (2019)								✓		
Hilmy et al. (2019)								✓		✓
Yang and Jiang (2019)								✓		
Pang et al. (2019)								✓		
Yao et al. (2020)					✓			✓		
Le et al. (2020)			✓						✓	
Thread recommender										
Yang, Piergallini et al. (2014)	✓	✓	✓		✓					
Yang, Adamson et al. (2014)	✓		✓							
Mi and Faltings (2016a)			✓			✓				
Mi and Faltings (2016b)			✓							
Mi and Faltings (2017)			✓							
Babinec and Srba (2017)		✓			✓					
Macina et al. (2017)					✓					
Lan et al. (2019)	✓		✓		✓					
Yang, Shang et al. (2014)					✓					
Chen and Epp (2020)		✓	✓				✓	✓		
Zhang et al. (2020)	✓		✓	✓						
Peer recommender										
Labarthe et al. (2016)										✓
Bouchet, Labarthe, Yacef et al. (2017)		✓			✓					
Prabhakar et al. (2017)					✓					
Bouchet, Labarthe, Bachelet et al. (2017)										✓
Potts et al. (2018)					✓					
Learning element recommender										
Elbadrawy et al. (2016)		✓			✓					
Pardos et al. (2017)					✓					
Zhao et al. (2018)										✓
Bhatt et al. (2018)										✓
Zhang, Huang et al. (2019)		✓		✓			✓			
Zhang, Zhu et al. (2019)		✓								
Campos et al. (2020)					✓					
Gong et al. (2020)					✓			✓		
Liu and Li (2020)				✓				✓		

Zhang, Hao et al. (2019) designed a hierarchical algorithm to identify noisy courses that contribute little or no interest of the user. This algorithm updates the user profile by removing noisy courses. Pang et al. (2019) used pre-requisites of courses and learning path of learners to recommend related courses to users. Yao et al. (2020) used course skills and description to cluster related courses and recommend them to users. They used LDA and page ranking algorithms for recommendations. Malgonde et al. (2020) designed an agent based two sided recommender system that can recommend different resources to students and to universities.

Some research reported comparative analyses. Piao and Breslin (2016) analysed the effect of different user modelling strategies (education field, job and skills) in a cold start situation and found that skill based modelling provided better results. Boratto et al. (2019) analysed biases related to course popularity, catalog coverage, and course category popularity. This research opened another area of research of eliminating bias from RS due to popularity of the items.

Most of the literature focuses on the design and implementation of course recommender systems considering different characteristics of learner and course in recommendation problem. Some recommender systems recommend courses on the basis of the behaviour of the learner, some on demographics and some combine both. The focus of this work is mostly the design of course recommender systems that recommend courses that match with users' interests or their actions, however in education the focus should also be given to the results of these recommender systems. Evaluation methods should include measurement of effects of these recommender systems on performance of students in MOOCs.

4.2. Thread recommender

In the design and implementation of thread RS, CF and CBF based methods are used for recommendations. Yang, Piergallini et al. (2014) designed an RS that utilizes student behaviour and interaction history for thread recommendation. Later (Yang, Shang et al., 2014) introduced sub modularity to decrease the complexity of their RS.

Agrawal et al. (2015) designed a video clip RS that classifies forum posts on the basis of sentiment, urgency and information variables then detects information confusion through information retrieval techniques. This system recommends related clips of lecture videos in order to help the student. Macina et al. (2017) considered load balancing and individual capabilities while recommending different threads for learners to answer. Zhang et al. (2020) designed a session based recommender system that considers all preferences of users by considering all the threads visited by a user and also takes into account threads visited by the learner in the current session. This system uses neural network to model the profile of the user.

Recommender systems can compute recommendations either offline/batch or online. Offline/batch recommender systems use a dataset containing the description of items and user's history to build a model which is used for recommendations (Khalid et al., 2017). On the other hand, online recommender systems do not require training for new items or users because online recommender systems only consider the current context of users for recommendations. Mi and Faltings (2016a, 2016b, 2017) developed an online thread RS that is based on a context tree. This system considers the current activities of learners for recommendation generation in the form of a sequence prediction problem. Their system adapts with the changes in user preferences.

Due to existing thread recommender systems in other areas (i.e social networks) work on thread recommender systems in MOOCs is very advanced. We found online thread recommender system, comparative studies on effectiveness of threads and thread recommenders with different constraints. There needs to be work on ways to encourage students to participate in threads because this increases engagement of students in the course which can of mutual benefit to all students in the course.

4.3. Peer recommender

Most of the work on peer RS is comparative studies and the impact of these systems on student progress. A reason for this trend could be the availability of peer recommender systems designed for social interaction websites which can also be utilized in MOOCs (Potts et al., 2018). Through a controlled study (Labarthe et al., 2016) observed the effect of peer RS on student grades, commitment and engagement in MOOCs and their results showed that peer recommender systems enhance student performance. The same group of authors (Bouchet, Labarthe, Yacef et al., 2017) then compared different peer RSs and evaluated their results which showed that socio-demographic based peer recommender systems have better results than course progress and random RS. When investigating the reasons for using peer RS, Bouchet, Labarthe, Bachelet et al. (2017) found that students who are not satisfied with other means of social interactions use these, but that they use them to share their emotions rather than to help each other in course progress.

Potts et al. (2018) and Prabhakar et al. (2017) designed a reciprocal peer RS that recommends related peers to students in a similar way to RSs used in dating websites. However, the systems in MOOCs used learning history and knowledge gaps of students for generating recommendations.

4.4. Learning element recommender

Work on the implementation of learning elements RSs began in 2016. We found one comparative study (Kopeinik et al., 2016) of different algorithm techniques and results from this study showed that the performance of any RS in MOOCs depends on characteristics of the dataset.

Pardos et al. (2017) designed and implemented a next page recommender system based on recurrent neural network. They used user click stream and time spent on a page to predict the next page that the user will visit and so recommend that page to the user. This was the first study we found that recommended learning resources but subsequently there has been increasing interest in this area of RSs for MOOCs. Liu and Li (2019) designed a learning path RS based on learning networks in which they made a course network and a learner network. To improve the quality of recommendation they divided learners in three types and recommended to them according to their type, for example, new learner. This work involved a complex network technology to use features of courses and learners. Jiang and Pardos (2019) used deep learning to design a next page RS that recommends pages to learners based on the goals of the learner. This system utilizes past actions to recommend future actions to give personalized recommendations to every user. Zhang, Zhu et al. (2019) designed a course guideline for learners based on their learning style, course knowledge and goals. In addition, they considered the student's engagement style while generating recommendations.

Bhatt et al. (2018) and Zhao et al. (2018) designed a video RS. They used topic modelling for modelling of videos and sequential pattern mining to find relationships between different topics of the videos and then used them to recommend related videos to learners in sequence and used CBF for their RS design. Subramanian et al. (2019) designed a chatbot for software engineers that uses natural language processing and content matching to recommend related resources to users. Liu and Li (2020) designed a recommender system that maintains course and learner networks and establishes relationships between different learners and courses based on their history in those courses, i.e. two courses will be connected if they are registered by the same user.

For the design and implementation of learning element recommender systems Deep learning, neural network and machine learning are mostly used. We did not find any online recommender system for learning elements that considers the current context of the learner and recommends learning resources with changes in the context of the learner. We also did not find any learning element recommender system that has used traditional recommendation techniques (i.e CF, CBf) for recommendation although these techniques are widely used for recommendation in different other areas (i.e Movies, video, etc.) .

4.5. Teacher support recommender

We found only one work related to a teacher support RS. Zhou et al. (2015) designed a user-friendly interface, collaborative mobile phone application that allows students to access their own performance and for them to provide ideas to improve the delivery of course content. These ideas are rated by their peers. In this way a student can view their own performance in the course and the teacher can receive feedback from their students and utilize their ideas to improve the delivery and content of their course. A MOOC course was used to evaluate the results of this RS.

Teacher support recommender systems can help teachers in designing and changing their MOOCs according to changes in the requirements of students and industries. These recommender systems could also decrease drop out rates in MOOCs.

5. Conclusion

The use of RSs in MOOCs has exciting opportunities to increase the popularity of MOOCs and to improve the learners' experience. Research to date has mostly focused on the implementation of course RSs in MOOCs, which was the most prolific research line throughout the period.

From 2012 to 2016, researchers focused on the implementation of course, peer and thread RSs, for which they mostly used CF, CBF and Hybrid techniques. Post 2016, there was an increasing focus on the implementation of learning element RSs and the use of neural network, deep learning and data mining, and in 2019 there was a larger number of papers on learning element RSs compared with thread RSs. Researchers started to focus on learners and to use their learning habits to recommend different learning resources to them.

In the early literature, researchers mostly used learner demographics, course characteristics for recommendation generation but from 2016 there has been an increasing trend in the usage of learner activities and learning style to design RSs. For learning element RSs, there has been an increase in designing next page RSs and mostly hybrid systems are used for thread RS with the use of sequential pattern mining showing promising results.

5.1. Gaps and future directions

Khalid et al. (2020) classified the types of recommender systems used in MOOCs into seven themes based on the area of a MOOC for which the RS is designed: course recommender, learning elements recommender, peer recommender, thread recommender, student performance recommender, MOOC provider/teachers' recommender and others. Work evaluating RSs that have been implemented in MOOCs has only reported on five areas (i.e Course, Thread, Peer, Learning Element, Teacher) with only one paper on teacher recommender systems. The remaining two areas of student performance and others have so far been ignored in terms of the evaluation of recommender systems implemented in MOOCs. Student performance recommender involves jobs, grades, student difficulty based, student dropout, work plan, and paid task recommenders. Others category involves improved and personalized MOOCs, adaptive content, and special user recommender systems. A further gap we found in the literature is the absence of any work on RSs for MOOC providers which leaves areas for which RSs can be developed.

Researchers have mostly used CF and CBF recommender systems for the design of recommender systems but most have not discussed their algorithms in the context of recommender system problems related to these techniques, such as the cold start problem, grey sheep users, outliers and data sparsity. There needs to be a focus on the design of a practical RSs for MOOCs while considering the above problems in traditional RSs.

There is an increasing trend in the of use of machine learning and neural networks for the design and implementation of learning element RSs. A rationale behind this trend could be the urge to use maximum features and behaviour of users to generate recommendations for which neural network is a better option. But these techniques can also be used for the design of RSs for thread and course recommender systems.

Different researchers have used different datasets for the evaluation of their RS because there is no standardized dataset available which makes bench-marking difficult. The datasets used are predominantly from the discipline of Computer Science despite there being MOOCs for other subject areas. There is only one rating-based publicly available MOOC dataset (Dessì et al., 2018) which contains only course teacher and rating data. There needs to be a complete dataset containing data from all areas of MOOCs where recommender systems can be applied. This will help in the analysis of different recommender systems and bench marking of results. There is also the need for data management tools for MOOC datasets. Some researchers have mentioned tools such as DataShop (Stamper et al., 2011), and MOOCdb (Veeramachaneni et al., 2014) that can be used while working on the immense dataset of MOOCs.

In addition, the types of MOOCs, for example, xMOOC and cMOOCs have not been considered nor discussed although each type has different characteristics which can influence the design of RSs. We have observed from the type of dataset that is used for evaluation of the recommender system that most recommender systems are designed for extended MOOCs (xMOOCs) but the type of MOOC has not been specifically mentioned by many authors. Precision and recall have predominantly been used to evaluate the performance of the recommender system but scalability and time complexity are two important factors that have not yet been considered researchers. We also observed an increase in the use of nDCG to measure the quality of ranking of the recommendation list.

We found only one online recommender system that can incorporate current user actions for thread recommendations. Online recommender systems should be designed for next page and learning element recommender systems to incorporate current user actions while recommending different resources to them.

MOOCs data maintain the profile of learners that contains characteristics of learners as well as their behavioural history. MOOCs also contain course details. A detailed comparative study could be performed that will observe and analyse the importance of different characteristics of learners and courses while computing recommendations.

The objective of RSs in MOOCs is to recommend items to learners according to their interests, skill level, background education and market requirements, so it is important to measure the accuracy and quality of recommended items. It is also important to observe the effect of the RS on the learners and providers. An effective recommender system for learners should help them in accomplishing their goals. Evaluation of RSs in MOOCs should focus on quantitative as well qualitative measurement of the recommendations. For qualitative evaluation different techniques such as surveys, interviews, questionnaires can be used to obtain feedback from learners, teachers and providers of the MOOCs. These techniques have been used by some researchers to carry out qualitative evaluation of their recommender systems and in the future, it can be an area to define techniques that can be used for qualitative evaluation of recommender systems in MOOCs.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This research work is supported by School of engineering and computer science, Victoria University of Wellington, New Zealand.

References

Agrawal, A., Venkatraman, J., Leonard, S., & Paepcke, A. (2015). YouEDU: Addressing confusion in MOOC discussion forums by recommending instructional video clips. In *EDM*.

Agrebi, M., Sendi, M., & Abed, M. (2019). Deep reinforcement learning for personalized recommendation of distance learning. In *World conference on information systems and technologies*. Springer, http://dx.doi.org/10.1007/978-3-030-16184-2_57.

Ahera, S. B., & Lobo, L. (2013). Combination of machine learning algorithms for recommendation of courses in E-learning system based on historical data. *Knowledge-Based Systems*, *51*, 1–14. http://dx.doi.org/10.1016/j.knosys.2013.04.015, URL: http://www.sciencedirect.com/science/article/pii/S0950705113001275.

Apaza, R. G., Cervantes, E. V., Quispe, L. C., & Luna, J. O. (2014). Online courses recommendation based on LDA. In *International conference on information management and big data, Vol. 1318* (pp. 42–48). SIMBig.

Babinec, P., & Srba, I. (2017). Education-specific tag recommendation in CQA systems. In *Adjunct publication of the 25th conference on user modeling, adaptation and personalization* (pp. 281–286). 3099081: ACM, http://dx.doi.org/10.1145/3099023.3099081.

Bhatt, C., Cooper, M., & Zhao, J. (2018). SeqSense: Video recommendation using topic sequence mining. In *MultiMedia modeling, International conference on multimedia modeling* (pp. 252–263). Springer International Publishing.

Boratto, L., Fenu, G., & Marras, M. (2019). The effect of algorithmic bias on recommender systems for massive open online courses. In *Advances in information retrieval* (pp. 457–472). http://dx.doi.org/10.1007/978-3-030-15712-8_30.

Bouchet, F., Labarthe, H., Bachelet, R., & Yacef, K. (2017). Who wants to chat on a MOOC? Lessons from a peer recommender system. In *Digital education: Out to the world and back to the campus*, (pp. 150–159). Springer International Publishing.

Bouchet, F., Labarthe, H., Yacef, K., & Bachelet, R. (2017). Comparing peer recommendation strategies in a MOOC. In *Adjunct publication of the 25th conference on user modeling, adaptation and personalization* (pp. 129–134). 3099036: ACM, http://dx.doi.org/10.1145/3099023.3099036.

Campos, R., dos Santos, R. P., & Oliveira, J. (2020). *SBSI'20, A recommendation system based on knowledge gap identification in MOOCs ecosystems*. New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/3411564.3411572.

Chen, Z., & Epp, C. D. (2020). CSCLRec: Personalized recommendation of forum posts to support socio-collaborative learning. In *EDM*.

Dahdouh, K., Dakkak, A., Oughdir, L., & Ibriz, A. (2019). Large-scale e-learning recommender system based on spark and hadoop. *Journal of Big Data*, *6*(1), 2. http://dx.doi.org/10.1186/s40537-019-0169-4.

Dakhel, G. M., & Mahdavi, M. (2013). Providing an effective collaborative filtering algorithm based on distance measures and neighbors ' voting.

Dessì, D., Fenu, G., Marras, M., & Reforgiato Recupero, D. (2018). COCO: Semantic-enriched collection of online courses at scale with experimental use cases. In A. Rocha, H. Adeli, L. P. Reis, & S. Costanzo (Eds.), *Trends and advances in information systems and technologies* (pp. 1386–1396). Cham: Springer International Publishing.

Dhawal Shah (2020a). By the numbers: MOOCs during the pandemic. https://www.classcentral.com/report/mooc-stats-pandemic/. [Online; accessed 7-December-2020].

Dhawal Shah (2020b). Online degrees slowdown: A review of MOOC stats and trends in 2019. https://www.classcentral.com/report/moocs-stats-and-trends-2019/. [Online; accessed 29-March-2020].

Dhawal Shah (2020c). Online degrees slowdown: A review of MOOC stats and trends in 2020. https://www.classcentral.com/report/mooc-stats-2020. [Online; accessed 7-December-2020].

Downes, S. (2008). Places to go: Connectivism & connective knowledge. *Innovate: Journal of Online Education*, *5*(1).

Elbadrawy, A., Polyzou, A., Ren, Z., Sweeney, M., Karypis, G., & Rangwala, H. (2016). Predicting student performance using personalized analytics. *Computer*, *49*(4), 61–69. http://dx.doi.org/10.1109/MC.2016.119.

Fauzan, F., Nurjanah, D., & Rismala, R. (2020). Apriori association rule for course recommender system. *Indonesia Journal on Computing (Indo-JC)*, *5*(2), 1–16. http://dx.doi.org/10.21108/INDOJC.2020.5.2.434, URL: https://socj.telkomuniversity.ac.id/ojs/index.php/indojc/article/view/434.

Fazeli, S., Rajabi, E., Lezcano, L., Drachsler, H., & Sloep, P. (2016). Supporting users of open online courses with recommendations: An algorithmic study. In *IEEE 16th international conference on advanced learning technologies (ICALT)* (pp. 423–427). http://dx.doi.org/10.1109/ICALT.2016.119.

Garg, V., & Tiwari, R. (2016). Hybrid massive open online course (MOOC) recommendation system using machine learning. In *International conference on recent trends in engineering, science & technology - (ICRTEST 2016)* (pp. 1–5). http://dx.doi.org/10.1049/cp.2016.1479.

Gong, J., Wang, S., Wang, J., Feng, W., Peng, H., Tang, J., & Yu, P. S. (2020). *SIGIR '20, Attentional graph convolutional networks for knowledge concept recommendation in MOOCs in a heterogeneous view* (pp. 79–88). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/3397271.3401057.

Gope, J., & Jain, S. K. (2017). A learning styles based recommender system prototype for edX courses. In *International conference on smart technologies for smart nation (SmartTechCon)* (pp. 414–419). http://dx.doi.org/10.1109/SmartTechCon.2017.8358407.

Hajri, H., Bourda, Y., & Popineau, F. (2018). Personalized recommendation of open educational resources in MOOCs. In *Computer supported education, International conference on computer supported education* (pp. 166–190). Springer International Publishing.

He, X., Liu, P., & Zhang, W. (2017). Design and implementation of a unified mooc recommendation system for social work major: Experiences and lessons. In *IEEE international conference on computational science and engineering (CSE) and IEEE international conference on embedded and ubiquitous computing (EUC), Vol. 1* (pp. 219–223). http://dx.doi.org/10.1109/CSE-EUC.2017.46.

Hilmy, S., De Silva, T., Pathirana, S., Kodagoda, N., & Suriyawansa, K. (2019). MOOCs recommender based on user preference, learning styles and forum activity. In *2019 international conference on advancements in computing (ICAC)* (pp. 180–185). http://dx.doi.org/10.1109/ICAC49085.2019.9103376.

Hou, Y., Zhou, P., Wang, T., Yu, L., Hu, Y., & Wu, D. (2016). Context-aware online learning for course recommendation of MOOC big data. arXiv, abs/1610.03147.

Hou, Y., Zhou, P., Xu, J., & Wu, D. O. (2018). Course recommendation of MOOC with big data support: A contextual online learning approach. In *IEEE conference on computer communications workshops (INFOCOM WKSHPS)* (pp. 106–111). http://dx.doi.org/10.1109/INFCOMW.2018.8406936.

Huang, P.-S., He, X., Gao, J., Deng, L., Acero, A., & Heck, L. (2013). *CIKM '13, Learning deep structured semantic models for web search using clickthrough data* (pp. 2333–2338). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/2505515.2505665.

Jain, H., & Anika (2018). Applying data mining techniques for generating MOOCs recommendations on the basis of learners online activity. In *IEEE 6th international conference on MOOCs, innovation and technology in education (MITE)* (pp. 6–13). http://dx.doi.org/10.1109/MITE.2018.8747056.

Jiang, W., & Pardos, Z. A. (2019). Time slice imputation for personalized goal-based recommendation in higher education. In *Proceedings of the 13th ACM conference on recommender systems* (pp. 506–510). 3347030: ACM, http://dx.doi.org/10.1145/3298689.3347030.

Jing, X., & Tang, J. (2017). Guess you like: course recommendation in MOOCs. In *Proceedings of the international conference on web intelligence* (pp. 783–789). 3106478: ACM, http://dx.doi.org/10.1145/3106426.3106478.

Jo, Y., Tomar, G., Ferschke, O., Rosé, C. P., & Gašević, D. (2016). Expediting support for social learning with behavior modeling. In *9th international conference on educational data mining*.

Kang, Y. (2014). An analysis on SPOC: Post-MOOC era of online education. *Tsinghua Journal of Education*, [ISSN: 1001-4519] *35*, 85–93.

Kardan, A. A., Narimani, A., & Ataiefard, F. (2017). A hybrid approach for thread recommendation in MOOC forums. *World Academy of Science, Engineering and Technology, International Journal of Social, Behavioral, Educational, Economic, Business and Industrial Engineering*, *11*, 2360–2366.

Khalid, A., Ghazanfar, M. A., Zahra, S., & Azam, M. A. (2017). Scalable and practical one-pass clustering algorithm for recommender system. *Intell. Data Anal.*, *21*, 279–310.

Khalid, A., Lundqvist, K., & Yates, A. (2020). Recommender systems for moocs: a systematic literature survey (january 1, 2012 – july 12, 2019). *The International Review of Research in Open and Distributed Learning*, *21*(4), 255–291. http://dx.doi.org/10.19173/irrodl.v21i4.4643, http://www.irrodl.org/index.php/irrodl/article/view/4643.

Kopeinik, S., Kowald, D., & Lex, E. (2016). Which algorithms suit which learning environments? A comparative study of recommender systems in TEL. In *Adaptive and adaptable learning, European conference on technology enhanced learning* (pp. 124–138). Springer International Publishing.

Labarthe, H., Bouchet, F., Bachelet, R., & Yacef, K. (2016). Does a peer recommender foster students' engagement in MOOCs? In *9th international conference on educational data mining* (pp. 418–423). URL: https://hal.archives-ouvertes.fr/hal-01376431.

Lan, A. S., Spencer, J. C., Chen, Z., Brinton, C. G., & Chiang, M. (2019). Personalized thread recommendation for MOOC discussion forums. In *Machine learning and knowledge discovery in databases*, *Proc. of the joint european conference on machine learning and knowledge discovery in databases* (pp. 725–740). Springer International Publishing, arXiv:1806.08468. doi:arXiv:1806.08468.

Le, T., Vo, V., Nguyen, K., & Le, B. (2020). Improving deep matrix factorization with normilized cros entropy loss function for graph-based MOOC recomendations. In *Big data analytics, data mining and computational intelligence 2020* (pp. 141–148). URL: https://www.cgv-conf.org/wp-content/uploads/2020/07/01_202011L017_F060.pdf.

Li, X., Xing, J., Wang, H., Zheng, L., Jia, S., & Wang, Q. (2018). A hybrid recommendation method based on feature for offline book personalization.

Liu, H., & Li, X. (2019). Learning path combination recommendation based on the learning networks. *Soft Computing*, http://dx.doi.org/10.1007/s00500-019-04205-x.

Liu, H., & Li, X. (2020). Learning path combination recommendation based on the learning networks. *Soft Computing*, *24*, http://dx.doi.org/10.1007/s00500-019-04205-x.

Liyanagunawardena, T., Adams, A., & Williams, S. (2014). MOOCs: A systematic study of the published literature 2008–2012. *Distance Education in China*, *3*, 5–16. http://dx.doi.org/10.19173/irrodl.v14i3.1455.

Lops, P., de Gemmis, M., & Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor (Eds.), *Recommender systems handbook* (pp. 73–105). Boston, MA: Springer US, http://dx.doi.org/10.1007/978-0-387-85820-3_3.

Lundqvist, K. Ø., & Warburton, S. (2019). Visualising learning pathways in MOOCs. In *2019 IEEE learning with MOOCS (LWMOOCS)* (pp. 185–190).

Macina, J., Srba, I., Williams, J. J., & Bielikova, M. (2017). Educational question routing in online student communities. In *Proceedings of the eleventh ACM conference on recommender systems* (pp. 47–55). 3109886: ACM, http://dx.doi.org/10.1145/3109859.3109886.

Malgonde, O., Zhang, H., Padmanabhan, B., & Limayem, M. (2020). Taming the complexity in search matching: Two-sided recommender systems on digital platforms. *MIS Quarterly*, *44*, 48–84. http://dx.doi.org/10.25300/MISQ/2020/14424.

Manouselis, N., Drachsler, H., Verbert, K., & Duval, E. (2012). *Recommender systems for learning*. Springer Publishing Company, Incorporated.

Mi, F., & Faltings, B. (2016a). Adapting to drifting preferences in recommendation. In *Neural information processing systems (NIPS 2016)* (p. 5). NIPS.

Mi, F., & Faltings, B. (2016b). Adaptive sequential recommendation using context trees. In *Proceedings of the twenty-fifth international joint conference on artificial intelligence* (pp. 4018–4019). 3061199: AAAI Press.

Mi, F., & Faltings, B. (2017). Adaptive sequential recommendation for discussion forums on MOOCs using context trees. In *10th international conference on educational data minings*. EDM.

Obeidat, R., Duwairi, R., & Al-Aiad, A. (2019). A collaborative recommendation system for online courses recommendations. In *2019 international conference on deep learning and machine learning in emerging applications (Deep-ML)* (pp. 49–54). http://dx.doi.org/10.1109/Deep-ML.2019.00018.

Panagiotis, S., & Dimitrios, M. (2018). Multi-modal matrix factorization with side information for recommending massive open online courses. *Expert Systems with Applications*, *118*, http://dx.doi.org/10.1016/j.eswa.2018.09.053.

Pang, Y., Liao, C., Tan, W., Wu, Y., & Zhou, C. (2018). Recommendation for MOOC with learner neighbors and learning series. In *Web information systems engineering – WISE 2018, International conference on web information systems engineering* (pp. 379–394). Springer International Publishing.

Pang, Y., Wang, N., Zhang, Y., Jin, Y., Ji, W., & Tan, W. (2019). Prerequisite-related MOOC recommendation on learning path locating. *Computational Social Networks*, *6*, http://dx.doi.org/10.1186/s40649-019-0065-2.

Pappano, L. (2012). The year of the MOOC. *The New York Times*, *2*(12), URL: https://www.nytimes.com/2012/11/04/education/edlife/massive-open-online-courses-are-multiplying-at-a-rapid-pace.html.

Pardos, Z. A., Tang, S., Davis, D., & Le, C. V. (2017). Enabling real-time adaptivity in MOOCs with a personalized next-step recommendation framework. In *Proceedings of the fourth (2017) ACM conference on learning @ scale* (pp. 23–32). 3051471: ACM, http://dx.doi.org/10.1145/3051457.3051471.

Piao, G., & Breslin, J. G. (2016). Analyzing MOOC entries of professionals on linkedin for user modeling and personalized MOOC recommendations. In *Proceedings of the 2016 conference on user modeling adaptation and personalization* (pp. 291–292). 2930264: ACM, http://dx.doi.org/10.1145/2930238.2930264.

Piedra, N., Chicaiza, J., López, J., & Caro, E. T. (2014). Supporting openness of MOOCs contents through of an OER and OCW framework based on linked data technologies. In *IEEE global engineering education conference* (pp. 1112–1117). http://dx.doi.org/10.1109/EDUCON.2014.6826249.

Potts, B. A., Khosravi, H., Reidsema, C., Bakharia, A., Belonogoff, M., & Fleming, M. (2018). Reciprocal peer recommendation for learning purposes. In *Proceedings of the 8th international conference on learning analytics and knowledge* (pp. 226–235). 3170400: ACM, http://dx.doi.org/10.1145/3170358.3170400.

Prabhakar, S., Spanakis, G., & Zaiane, O. (2017). Reciprocal recommender system for learners in massive open online courses (MOOCs). In *Advances in arning – ICWL 2017, International conference on web-based learning* (pp. 157–167). Springer International Publishing.

Rahma, B. D., & Koutheair, K. M. (2019). Towards a framework for building automatic recommendations of answers in MOOCs' discussion forums. In *2019 7th international conference on ICT accessibility (ICTA)* (pp. 1–6). http://dx.doi.org/10.1109/ICTA49490.2019.9144850.

Raval, U. R., & Jani, C. (2016). Implementing & improvisation of K-means clustering algorithm.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J. (1994). GroupLens: An open architecture for collaborative filtering of netnews. In *CSCW '94, Proceedings of the 1994 ACM conference on computer supported cooperative work* (pp. 175–186). New York, NY, USA: ACM, http://dx.doi.org/10.1145/192844.192905, URL: http://doi.acm.org/10.1145/192844.192905.

Ricci, F., Rokach, L., Shapira, B., & Kantor, P. B. (2010). *Recommender systems handbook* (p. 842). Springer-Verlag, http://dx.doi.org/10.1007/978-0-387-85820-3.

Sakboonyarat, S., & Tantatsanawong, P. (2019). Massive open online courses (MOOCs) recommendation modeling using deep learning. In *2019 23rd international computer science and engineering conference (ICSEC)* (pp. 275–280). http://dx.doi.org/10.1109/ICSEC47112.2019.8974770.

Sidi, L., & Klein, H. (2020). Neural network-based collaborative filtering for question sequencing.

Stamper, J. C., Koedinger, K. R., Baker, R. S. J. d., Skogsholm, A., Leber, B., Demi, S., Yu, S., & Spencer, D. (2011). DataShop: A data repository and analysis service for the learning science community (interactive event). In G. Biswas, S. Bull, J. Kay, & A. Mitrovic (Eds.), *Artificial intelligence in education* (p. 628). Berlin, Heidelberg: Springer Berlin Heidelberg.

Subramanian, V., Ramachandra, N., & Dubash, N. (2019). TutorBot: Contextual learning guide for software engineers. In *2019 IEEE/ACM 1st international workshop on bots in software engineering (BotSE)* (pp. 16–17). http://dx.doi.org/10.1109/BotSE.2019.00011.

Trirat, P., Noree, S., & Yi, M. (2020). IntelliMOOC: Intelligent online learning framework for MOOC platforms. In *EDM*.

Veeramachaneni, K., Halawa, S., Dernoncourt, F., O'Reilly, U.-M., Taylor, C., & Do, C. (2014). MOOCdb: Developing standards and systems to support MOOC data science. arXiv:1406.2015.

Wang, Y., Liang, B., Ji, W., ShiweiWang, & YiqiangChen (2017). An improved algorithm for personalized recommendation on MOOCs. *International Journal of Crowd Science*, *1*(3), 186–196. http://dx.doi.org/10.1108/IJCS-08-2017-0021.

Yang, D., Adamson, D., & Rosé, C. P. (2014). Question recommendation with constraints for massive open online courses. In *Proceedings of the 8th ACM conference on recommender systems* (pp. 49–56). 2645748: ACM, http://dx.doi.org/10.1145/2645710.2645748.

Yang, X., & Jiang, W. (2019). Dynamic online course recommendation based on course network and user network. In G. Wang, A. El Saddik, X. Lai, G. Martinez Perez, & K.-K. R. Choo (Eds.), *Smart city and informatization* (pp. 180–196). Singapore: Springer Singapore.

Yang, D., Piergallini, M., Howley, I., & Rose, C. P. (2014). Forum thread recommendation for massive open online courses. In *7th international conference on educational data mining*.

Yang, D., Shang, J., & Rosé, C. P. (2014). Constrained question recommendation in MOOCs via submodularity. In *Proceedings of the 23rd ACM international conference on conference on information and knowledge management* (pp. 1987–1990). 2662089: ACM, http://dx.doi.org/10.1145/2661829.2662089.

Yanhui, D., Dequan, W., Yongxin, Z., & Lin, L. (2015). A group recommender system for online course study. In *7th international conference on information technology in medicine and education (ITME)* (pp. 318–320). http://dx.doi.org/10.1109/ITME.2015.99.

Yao, W., Sun, H., & Hu, X. (2020). A novel search ranking method for MOOCs using unstructured course information. *Wireless Communications and Mobile Computing*, *2020*, 8813615:1–8813615:13. http://dx.doi.org/10.1155/2020/8813615.

Zhang, J., Hao, B., Chen, B., Li, C., Chen, H., & Sun, J. (2019). Hierarchical reinforcement learning for course recommendation in MOOCs. (pp. 435–442). http://dx.doi.org/10.1609/aaai.v33i01.3301435, URL: https://ojs.aaai.org/index.php/AAAI/article/view/3815,

Zhang, H., Huang, T., Lv, Z., Liu, S., & Yang, H. (2019). MOOCRC: A highly accurate resource recommendation model for use in MOOC environments. *Mobile Networks and Applications*, *24*(1), 34–46. http://dx.doi.org/10.1007/s11036-018-1131-y.

Zhang, H., Huang, T., Lv, Z., Liu, S., & Zhou, Z. (2018). MCRS: A course recommendation system for MOOCs. *Multimedia Tools and Applications*, *77*(6), 7051–7069. http://dx.doi.org/10.1007/s11042-017-4620-2.

Zhang, M., Liu, S., & Wang, Y. (2020). STR-SA: Session-based thread recommendation for online course forum with self-attention. In *2020 IEEE global engineering education conference (EDUCON)* (pp. 374–381). http://dx.doi.org/10.1109/EDUCON45650.2020.9125245.

Zhang, H., Yang, H., Huang, T., & Zhan, G. (2017). DBNCF: Personalized courses recommendation system based on DBN in MOOC environment. In *International symposium on educational technology* (pp. 106–108). http://dx.doi.org/10.1109/ISET.2017.33, URL: https://ieeexplore.ieee.org/document/8005400.

Zhang, M., Zhu, J., Wang, Z., & Chen, Y. (2019). Providing personalized learning guidance in MOOCs by multi-source data analysis. *World Wide Web*, *22*(3), 1189–1219. http://dx.doi.org/10.1007/s11280-018-0559-0.

Zhao, J., Bhatt, C., Cooper, M., & Shamma, D. A. (2018). Flexible learning with semantic visual exploration and sequence-based recommendation of MOOC videos. In *Proceedings of the 2018 CHI conference on human factors in computing systems* (pp. 1–13). 3173903: ACM, http://dx.doi.org/10.1145/3173574.3173903.

Zhou, M., Cliff, A., Krishnan, S., Nonnecke, B., Crittenden, C., Uchino, K., & Goldberg, K. (2015). M-CAFE 1.0: Motivating and prioritizing ongoing student feedback during MOOCs and large on-campus courses using collaborative filtering. In *Proceedings of the 16th annual conference on information technology education* (pp. 153–158). 2808020: ACM, http://dx.doi.org/10.1145/2808006.2808020.