Ontology-based E-learning Content Recommender System for Addressing the Pure Cold-start Problem

JEEVAMOL JOY, NISHA S. RAJ, and RENUMOL V. G.,
Cochin University of Science and Technology, India

E-learning recommender systems are gaining significance nowadays due to its ability to enhance the learning experience by providing tailor-made services based on learner preferences. A Personalized Learning Environment (PLE) that automatically adapts to learner characteristics such as learning styles and knowledge level can recommend appropriate learning resources that would favor the learning process and improve learning outcomes. The pure cold-start problem is a relevant issue in PLEs, which arises due to the lack of prior information about the new learner in the PLE to create appropriate recommendations. This article introduces a semantic framework based on ontology to address the pure cold-start problem in content recommenders. The ontology encapsulates the domain knowledge about the learners as well as Learning Objects (LOs). The semantic model that we built has been experimented with different combinations of the key learner parameters such as learning style, knowledge level, and background knowledge. The proposed framework utilizes these parameters to build natural learner groups from the learner ontology using SPARQL queries. The ontology holds 480 learners' data, 468 annotated learning objects with 5,600 learner ratings. A multivariate k-means clustering algorithm, an unsupervised machine learning technique for grouping similar data, is used to evaluate the learner similarity computation accuracy. The learner satisfaction achieved with the proposed model is measured based on the ratings given by the 40 participants of the experiments. From the evaluation perspective, it is evident that 79% of the learners are satisfied with the recommendations generated by the proposed model in pure cold-start condition.

16

CCS Concepts: • **Information systems → Information systems applications**; **Computing platforms**;

Additional Key Words and Phrases: Learner profile, learning object, personalized learning environment, pure cold-start problem, content recommenders, ontology, multivariate clustering

ACM Reference format:
Jeevamol Joy, Nisha S. Raj, and Renumol V. G. 2021. Ontology-based E-learning Content Recommender System for Addressing the Pure Cold-start Problem. *J. Data and Information Quality* 13, 3, Article 16 (April 2021), 27 pages.
https://doi.org/10.1145/3429251

1 INTRODUCTION

A learning system based on formalized teaching with electronic resources is known as e-learning [1]. E-learning systems are convenient and flexible, since the learning resources are available from

Authors' addresses: J. Joy, N. S. Raj, and Renumol V. G, Division of Information Technology, School of Engineering, Cochin University of Science and Technology, Kerala-682002, India; emails: {jeeva.loy, nisha.s.raj, renumolvg}@gmail.com.

1936-1955/2021/04-ART16 $15.00
https://doi.org/10.1145/3429251

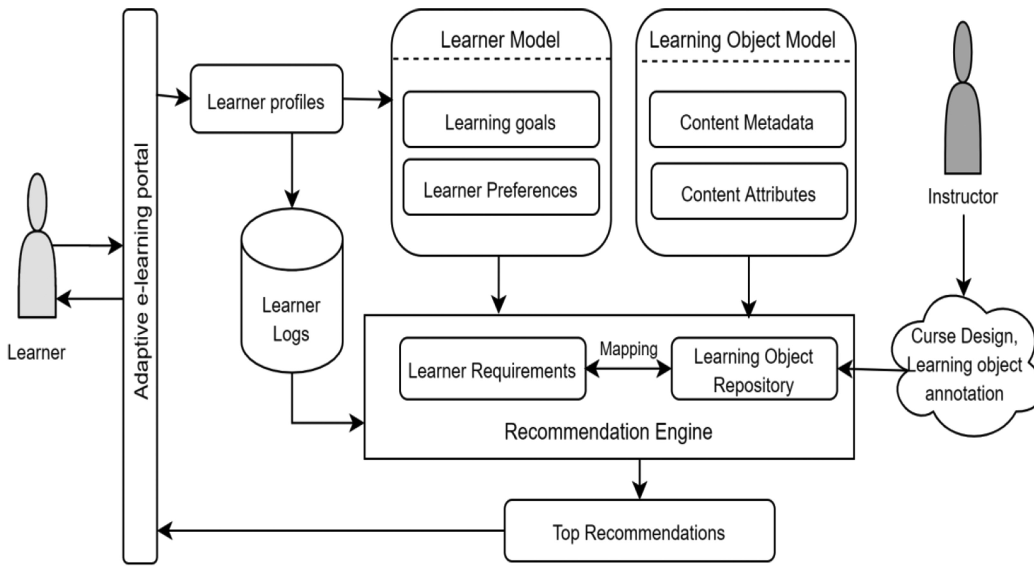

Fig. 1. The architecture of an e-learning content recommender system.

anywhere and at any time. The main advantage of using an e-learning platform is that it allows students to learn at their own pace. Maximizing knowledge retention is another most rewarding benefit of e-learning. It is capable of accommodating everyone's needs. Scalability and reduced cost are also advantages of these systems. E-learning provides expeditious lessons delivery, whereas traditional classrooms are often disturbed by mischievous elements [2]. E-Learning has many applications, such as web-based learning, computer-based learning, virtual classrooms, and digital collaboration. Content development and content delivery are two essential processes of any e-learning system. These processes required adequate analysis, planning, design, development, and delivery to develop a goal-oriented e-learning system [3]. The success of an e-learning system depends on the self-motivation of individuals using these systems.

In e-commerce, Recommender Systems (RSs) have emerged to deal with the information overload problem and recommend new items and products based on the users' preferences. The RSs use various data sources to build their recommendations [4–6]. These data sources define the characteristics of users, items, and their relationships. The data sources include users' demographic data, interests, preferences, and previous history. The standard recommendation techniques for predicting user preferences are collaborative filtering (CF), content-based filtering (CB), or a hybrid approach. The most popular RS design method is the CF approach, which uses the user's feedback to make relevant future recommendations [7, 8]. CB systems try to match the user profiles against item's description to generate recommendations [9]. Hybrid methods have evolved by combining CF and CB models to provide more accurate results [10]. The above three conventional techniques require users' historic ratings for finding similar user groups and making appropriate recommendations. In the e-learning domain, RSs are widely used for content recommendation. A content RS uses the above recommendation techniques to recommend personalized learning content. The general architecture of an e-learning content RS is shown in Figure 1.

One of the most known problems associated with RSs is the cold-start problem. Typically, two main classifications exist for this problem: new item cold-start problem and new user cold-start problem [11]. The new item cold-start problem occurs when a new item is introduced to the RS, because the ratings available for this item are zero or significantly less. This issue can be solved to a certain extent by giving prior ratings to the newly introduced item by the staff members. The new user cold-start problem occurs when a new user has been registered in the system [5]. When a new user becomes part of the system, the system has no information about the users' prior choices. But it is the responsibility of the system to start suggesting items for the new user also. Otherwise, it will negatively affect the RS's performance, and users may stop using the system due to its inability to provide meaningful recommendations. Thus, in the new user cold-start condition, the challenge is to recommend items to the new user without knowing their prior choices. Most of the research works conducted on new user cold-start issues address only the partial cold-start condition [6, 12, 13]. They assume that at least a few ratings/feedbacks are available for the new user. Whereas, in the pure cold-start condition, the number of user ratings available is zero, since they have not yet rated any items. Similar to other e-commerce domains, the pure cold-start problem occurs in the e-learning domain when the learner uses the Learning Management System (LMS) initially. The RS knows only the learner attributes such as demographic data, learning style, learners' background knowledge, and knowledge level. One way to predict the most suitable Learning Object (LO) with this information is to compute the learner profile for the new learner by grouping similar learners [14]. This work focuses on solving the pure cold-start problem that occurs in content recommenders in PLEs.

In a Personalized Learning Environment (PLE), the main focus is to understand and adapt to their learners' needs. Learners have different individual needs, objectives, and preferences that affect their learning processes [15]. Similarly, different learners have different characteristics regarding learners' background knowledge, learners' history, competency level, learning style, and learning activities. This difference in learner characteristics makes the recommendation of learning resources to a particular learner more difficult. One of the main challenges in such systems is that user interests, preferences, and needs are not fixed, but change over time. If learner profiles contain just static information, then this will lead to constraining the personalization process and recommending irrelevant services and items over a period of time [16]. Therefore, the focus of RSs in e-learning environments should be to generate accurate and personalized recommendations based on learners' specific demands and requirements. The existing PLEs exhibit cold-start problems and issues related to the mapping of learner data with LOs [17–19]. To solve these issues and improve the dynamicity of the PLEs, an appropriate learner/learning object model is essential [20]. In response to the problems mentioned above, we propose an ontology-based recommendation model to address the pure cold-start problem. In this study, the ontological knowledge is combined with the conventional collaborative filtering technique. The ontology provides the initial knowledge about the learner and learning objects. Since historical ratings of similar learners are used in the recommendation logic, CF is also a part of the proposed model. The implications of the proposed model in the e-learning domain include:

- The dropout of new learners from e-learning RSs in the early stage due to irrelevant recommendations can be avoided.
- The learning outcomes in PLEs can be improved by making use of learner characteristics and preferences.

This article is an extension of our previously published work [21], in which we have presented the general ontology model for the content recommendation in PLEs. This work aims to propose the following innovative deliverables:

- Ontology-based system framework for addressing the pure cold-start problem.
- Identification of the semantic processes required for learner grouping and mapping the learner data with LOs in pure cold-start condition.

The proposed framework is evaluated using a real learner dataset consisting of 480 learners' data and 468 annotated learning objects with 5,600 learner ratings. A multivariate clustering technique is utilized to evaluate the accuracy of the proposed model in learner similarity computation. The learner satisfaction with the proposed model in the cold-start condition is estimated based on the ratings given to the LOs by a set of 40 learners who participated in the experiments.

The rest of the article is organized as follows: A literature review is included, which mainly focuses on different studies conducted for addressing the cold-start problem, Ontology-Based (OB) content recommenders in PLEs, and learner/learning object modeling characteristics. The ontology building process and the attributes of the developed ontology are explained subsequently. After this, the semantic process for extracting learner groups from the ontology and the recommendation logic are explained. The details about the experiments conducted and the data evaluation procedures are explained further. The article is concluded with the discussion of the experimental results and the future works planned.

2 RELATED WORKS

This section provides a detailed overview of the related works from four areas. In the first subsection, a summary of various studies conducted to address the pure cold-start problem in RSs, and how the RSs address this problem is detailed. Second, how ontologies are utilized in e-learning RSs for addressing the drawbacks associated with conventional RSs are presented. In the third subsection, the existing studies emphasizing the importance of learner modeling in RSs and ontology-based learner profiles are elaborated. Finally, the different works conducted with ontologies for metadata annotation of LOs, especially with the IEEE LOM standard, are explained.

2.1 Addressing the Pure Cold-start Problem

In the literature, the pure cold-start problem is mentioned as a subtype of the cold-start problem. Despite being closely analyzed, both the problems should be addressed separately. The cold-start problem is related to generating meaningful recommendations with minimal historical data. The pure cold-start problem is related to creating appropriate recommendations without historical data about the users [22]. There are many different strategies for addressing the cold-start issue with minimum historical data. Still, in the case of the pure cold-start condition, there are only a few approaches to satisfy first-time users [23, 24]. Most of the practices described in the literature are not capable of dealing with the pure cold-start problem [25–29]. Computing user similarities using mapping functions with factorization models [30], Pearson correlation [31], PIP [32], and matrix row clusters to build a classification model [33] are few exceptions found in the literature to tackle the pure cold-start condition. The two main categories of RSs designed to deal with the pure cold-start problem discussed in the literature are knowledge-based RSs and social filtering RSs [34]. Knowledge-based RSs use the users' domain knowledge, which is usually collected from the user during their first interaction with the RS. The social filtering RSs exploit external information about users, such as social, demographic, and personal data.

In general, the above RSs use hybrid strategies to recommend items after modeling a user's profile with social, demographic, or personal data. Chikhaoui et al. [35] proposed a demographic filtering hybrid RS to overcome the pure cold-start problem by correlating new users to active ones through their demographic information, such as age, gender, and professional occupation. In their study, users were categorized (clusters) using the nearest neighbor technique, with each cluster holding, users sharing similar demographic characteristics. In their study, users were clustered using the nearest neighbor technique, each cluster holding users with similar demographic characteristics. In another study conducted by Safoury and Salah [6], they also suggested utilizing users' demographic similarity to alleviate the pure cold-start problems in RSs. They have used rating history along with demographic data while selecting the neighborhood for the target user. The experimentations are conducted using the existing MovieLens dataset, not with real learner data. Rosli et al. [36] presented a hybrid approach that uses information obtained from social networks to improve the pure cold-start problem in RSs. The authors used the comments posted on Facebook pages to calculate the similarity measure and get the preliminary user profile. They have utilized the clustering technique to group similar learners based on their opinion on social media. The prediction coverage results show that their recommendation model is not acceptable in pure cold-start condition. Even if many advantages are obtained by demographic/social–based RSs, they are not commonly used in e-commerce scenarios, since many users are interested in buying items without providing any social or demographic information [37].

Different methodologies are adopted for addressing the cold-start problem in RSs. Most of the existing works combine conventional recommendation approaches with data mining/machine learning techniques. Moreno et al. [38] proposed a methodology that combines data mining techniques with semantic data to overcome the cold-start problem. In their study, several data mining algorithms were applied in the MovieLens dataset to select the best classifier in the cold-start condition. Sobhanam and Mariappan [39] combined two existing approaches to get better results in cold-start recommendations. They have used the association rule technique to expand the user profile and clustering technique to generate the final recommendations. Tarus et al. [40] proposed combining ontology with sequential pattern mining to solve this issue. In their methodology, the ontological domain knowledge helps to mitigate the new user cold-start problem. The existing studies discussed here had tried to solve the cold-start problem by combining conventional recommendation techniques such as CF and CB with other methods such as data mining, machine learning, and ontologies. The main drawback with knowledge-based RS is that the quality of recommendations mostly depends on users' information and sometimes users fail to define their preferences [41] clearly. The ontology model that we have created can update the learner profile dynamically once the learner starts using the LMS. By exploiting the updated learner profile of existing learners, the RSs can solve the cold-start problem largely.

2.2 Ontology-based Content Recommenders

Ontology-based RSs have evolved to overcome the limitations associated with conventional RSs. In content recommendation, ontology is a way to model learners and learning resources. Many research works have been conducted on the topic "how personalization can be achieved using ontologies."

Ontology is described as the conceptual specification of a particular domain of interest. Typically, OB recommender systems are knowledge-based systems that use ontology to represent knowledge about the items and users in the recommendation process [42]. In the context of e-learning, these systems use ontology knowledge about the learner and the learning resources to map a learner to the relevant learning resources. Ontologies play an essential role in knowledge representation, sharing, and reuse [43]. They are beneficial to define a semantic model of the data combined with its associated domain knowledge. Different types of semantic knowledge can be linked using ontologies, and this can be used in formulating strategies for searching data. Previous studies have shown that aggregation of ontology domain knowledge about the learner and learning resources improves the accuracy, quality of recommendations, and alleviates other drawbacks such as cold-start and rating sparsity problems associated with conventional recommendation techniques [44, 45]. Ontologies can help solve these issues by providing initial knowledge about users and their interest domains.

Ontologies are widely used in e-learning RSs. Tarus et al. [20] have presented a detailed review of several OB recommender systems in the e-learning domain. The authors' main finding is that domain ontology is used in the majority of the ontology-based RSs, and web ontology language is used for learner and LO modeling. The personalization framework developed by Ouf et al. [46] for content recommendation uses four ontologies to represent the learner, the learning object, the learning activities, and the teaching method. In their framework, the personalization dimension is enhanced by using semantic rules over these ontologies. The authors validated the impact of using ontologies in PLEs. Pukkhem [47] designed a LO recommendation system with multiple agents, in which ontologies are utilized for creating the agents. Their framework uses ontologies to interpret and process LOs in the RS. Their study's drawback is that the semantic rules are built based on the learner attribute "learning style" alone. Tarus et al. [40] developed a hybrid recommendation strategy for the content recommendation in the e-learning domain. The authors implemented ontology over the learners and learning resources for making relevant recommendations. They have integrated a sequential pattern mining technique with the ontology model to identify the learner's historical sequential pattern from weblogs. The results show that the usage of ontological domain knowledge improves the performance of the RS in cold-start conditions. The drawback of their study is that the learner's learning style parameter is extracted through the questionnaire method. Bouihi and Bahaj [48] designed a semantic layer to be integrated into the current e-learning platforms and pointed out the benefits of semantic layer integration. Their methodology provides a coherent model for storing and processing context information in the semantic layer after building the e-learning ontology. Nevertheless, their architecture does not address how the cold-start problem existing in PLEs is mitigated with context information. Klašnja-Milićević et al. [14] proposed a personalized e-learning system that can automatically adapt to learners' interests, habits, and knowledge levels using domain ontology. Their methodology adopts the data mining technique (Aprioriall algorithm) to identify learners' sequential patterns and the CF technique to LO recommendation. Their model is not suitable in pure cold-start condition, since it relies mainly on historical ratings of learners. Saleena and Srivatsa [49] designed an adaptive learning system by utilizing a fuzzy domain ontology and domain expert's ontology. The cross ontology and concept similarity measures incorporated in the proposed model increased the RS's accuracy and performance. Still, they are not specifying the performance of the learning system in the cold-start condition.

Ontologies are extensively used in RSs for knowledge representation due to its wide range of benefits [50]. Fraihat and Shambour [51] utilized the semantic relation between learning activities and learning objects to solve the RS's cold-start problem. Their recommendation strategy uses ontology to classify LOs based on concepts. Their model's limitation is that the model is based on learner ratings and, therefore, useful in partial cold-start condition alone. Harrathi et al. [52] attempted to solve the cold-start problem using a hybridized approach using ontologies. Their proposed model utilizes Bloom's taxonomy to represent the learning attributes. The model needs to be tested in cold-start conditions to measure personalization performance. Obeid et al. [53] proposed a recommendation model that considers students' skills, interests, and preferences to address the cold-start problem. In their model, machine learning techniques are used to learn the student profile. Their system's drawback is that the performance of the model degrades if the students fail to complete the profile with personal information. In general, Ontology-based content recommenders aggregate the knowledge about the learner and learning objects, and this knowledge is exploited in the recommendation process [54]. The quality of recommendations and the problems associated with conventional RSs (cold-start and rating sparsity) can be improved by using OB recommenders [55]. The accuracy and completeness of knowledge acquired in the domain ontology affect the effectiveness of ontology-based recommender systems. Therefore, the effective modeling of learner and learning objects in the ontology is crucial in OB recommender systems. In the following section, the existing literature that features learner and learning object modeling is explained.

2.3 Learner Modeling

Many studies have been conducted on OB user profiles for constructing personalized learning models [56–59]. Ontologies have proved to be successful in expressing information in machine-processable representation. The learner profile is practically the standard representation of learner's data that can be gathered in two ways: directly from the learner or by analyzing his/her behavior through an LMS. If the details are gathered directly from the learner, then the profile made is called an explicit or static profile. Whereas if this information is collected by observing the learner's behavior in an LMS, then the profile created is known as the implicit or dynamic profile. A good learner profile can be effortlessly adjusted for every learner according to his/her preferences. Learner models play a vital role in the personalization of e-learning systems. In a PLE, the most suitable learning resources are recommended to learners based on their learning characteristics [60]. RSs use different features, such as learners' prior knowledge, motivation level, objectives, and preferences for personalization [61]. The learning style is also considered in many adaptive e-learning systems to model individual differences of learners for personalization purposes [62]. At a single knowledge point, the learner can choose to learn from theoretical or practical pedagogies and select from beginner to expert level of materials as per the learner requirement.

Sheeba and Krishnan [63] proposed an ontology-based learner profile approach to achieve effective information retrieval. The sophisticated representation of static and dynamic learner characteristics is an advantage of their learner model. Their methodology includes a decision tree classifier to classify the learning styles automatically. The experimental evaluation reveals enhanced performance with an ontological representation of the user profile. Liang et al. [64] developed a document RS that adopts a semantic-expansion approach that includes "is-a" and "non-is-a" relationships for concept mapping. Their approach's limitation is that it requires the user to provide relevant feedback to build the browsing profile. Nafea et al. [65] proposed an adaptive learner profile based on learning styles by analyzing the learning patterns through an LMS. The theoretical base for building their learner profile is the Felder-Silverman Learning Style Model (FSLSM) and Myers-Briggs Type Indicator (MBTI) theory. Chen et al. [66] considered background knowledge, learner history, and learning style for learner modeling. The limitation of their learner profile is its static nature. The learner model proposed by Klašnja-Milićević et al. in a personalized e-learning system considered the learner interests, habits, and knowledge levels [14]. Their approach is capable of recognizing different patterns of learners' practices through mining server logs. Tarus et al. [40] included the learners' learning style and knowledge level in their learner model. Since the required data for building the learner profile is collected through questionnaires alone, learner modeling's dynamicity is lacking in their recommendation model.

We have considered all the relevant learning parameters, such as learners' learning styles, knowledge level, and qualifications for building our learner model. Along with these learner parameters, each learner's learning path is stored in the ontology to provide dynamicity in the learner profile. We have selected the Felder-Silverman Learning Style Model (FSLSM) [67], because this model brings together different elements from the models by Kolb (1981), Pask (1976), and Myers-Briggs (1962). Also, a lot of research has been conducted using the FSLSM to provide personalization in the e-learning domain.

2.4 Learning Object Modeling

A learning object is defined as any entity, digital or non-digital, that can be used for learning, education, or training [68]. The learning objects should be kept in a standardized manner so it can be easily stored, accessed, and retrieved in the PLE. LOs are widely purposed and reused as a meaningful and effective way of creating content for e-learning, especially within learning and course management systems. Metadata annotations for learning resources provided by LO repositories play a crucial role, since they essentially convey machine-readable descriptions for the LOs. Ontologies are widely used for metadata modeling of LOs (especially with the IEEE LOM standard) in e-learning applications [69]. Al-Yahya et al. [70] discussed how ontologies help extract relevant information when the information is retrieved based on the learning objects' semantics. Raju and Ahmed [71] describe a model for representing a learning object repository using ontologies. The developed domain ontology stores annotated LOs, resulting in a discoverable and reusable learning object. Sosnovsky et al. [72] used ontology in their personalized recommender system for storing adaptively annotated learning material. The recommended learning material's annotation level is computed as a weighted aggregate of knowledge levels for all concepts mapped into the topic. Cakula and Sedleniece [73] tried to identify the overlapping points between e-learning phases and knowledge management to improve personalized learning materials delivery. By using adequate ontology and metadata methods, the e-learning resources can be easily customized and personalization can be achieved. In most OB content recommenders, the learning materials are also stored in ontologies to efficiently map with the learner characteristics [74–77].

Existing literature shows that different studies have been conducted with ontologies to model the learner and LO metadata in PLEs. But not much research has been done by combining the learner and learning objects characteristics as a single model for the content recommendation in PLEs. Though several different studies are conducted in e-learning content RSs, a more reliable approach handling drawbacks like cold-start issues is yet to be developed. The significant gaps identified in the existing literature are:

- Most of the existing works handle partial cold-start problem only.
- Minimal research works only have been conducted to address the pure cold-start problem in content recommenders using ontologies.
- Most of the studies reported in the literature have experimented with existing datasets (MovieLens, GroupLens, and Netflix) in cold-start conditions, and experimentations with the actual learners lack in this area.

In this study, we are trying to fill these gaps by proposing an OB recommender system that can address the pure cold-start problem and improve the recommendations' quality. The details of the proposed system framework are explained in the subsequent section.

3 THE PROPOSED SYSTEM FRAMEWORK

The primary goal of the proposed system is to address the pure cold-start problem in e-learning content recommender systems. If the RS can assign a new learner to the most similar group of learners, then it can create relevant recommendations for the new learner. Similarly, if the RS can map between the learner characteristics with LO attributes, then the quality of recommendations can be improved further. The main components of the proposed framework are learner interfaces, ontologies, learner grouping unit, and an OB recommender, as shown in Figure 2.

The initial learner attributes have to be collected directly from the learner when the learner starts using the LMS initially. The learner interfaces were developed to aid this purpose. The learner interfaces play an essential role in this system, because the learner interacts with the LMS through the learner interfaces only. All the prominent learner attributes such as demographic data, learning style (through a questionnaire), knowledge level, and background knowledge are collected through these interfaces. Other than this, a learner can change the learner profile information and choose any LO from the system using this interface. When a new learner selects the

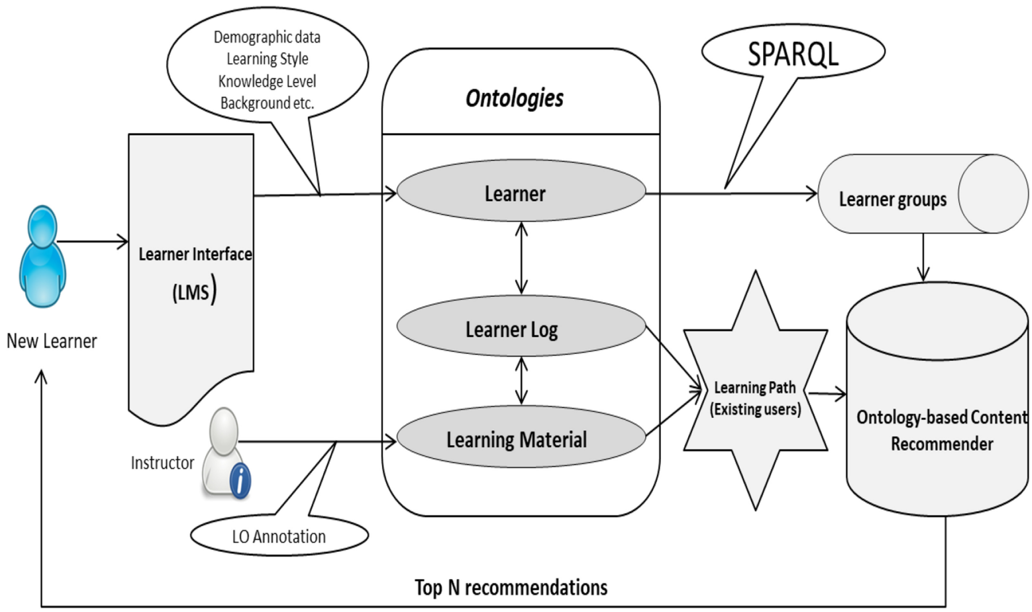

Fig. 2. The system framework for addressing the pure cold-start problem in content recommenders.

learning topic and requests the LO, the system runs the predefined semantic query on the ontology and returns appropriate recommendations. The learner can start learning from this point. Ontologies are built to store the data collected from the learner, learning object metadata (annotated by the instructor), and each learner's learning path. The details about the ontology building processes are included in the subsequent section, "Building Ontologies." In the case of a new learner, natural learner groups are generated by running SPARQL queries against *Learner* ontology. The learning history of the existing learners, who are included in the learner group extracted by running the SPARQL query, can be found from the *LearnerLog* and *Learning Material* sub-ontologies. The acquired learner groups and learning path information are inputted to the OB content recommender to produce the top N recommendations for the new learner. The steps involved in generating top N recommendations in pure cold-start condition are:

- Creating domain ontologies for storing learner, learning material, and learner log data.
- Semantic rule mapping between the learner and the learning materials.
- Extracting similar learner groups using SPARQL queries.
- Generating top N recommendation list of learning objects by the ontology-based recommendation engine by using collaborative filtering technique and semantic rule mapping.

In this work, ontologies are mainly utilized for knowledge representation. The ontology stores domain knowledge about the learner and LO, along with their relationships. Semantic rules are built to match between learners and LO characteristics in pure cold-start condition. Along with this, the learning paths (includes ratings) of existing learners are utilized to generate the cold-start recommendations. The ontology learns how a learner progresses through the LMS, and this information is used to create recommendations to the target learner in cold-start conditions. This article addresses the pure cold-start problem alone, by semantic rule mapping and utilizing a collaborative filtering technique.

Table 1. FSLSM Dimensions

Continuum	Learning Style Dimensions
Active/Reflective	How a learner prefers to process information.
Sensitive/Intuitive	How a learner prefers to perceive or take in information.
Visual/Verbal	How a learner prefers information to be presented.
Global/Sequential	How a learner prefers to organize and progress toward understanding information.

4 BUILDING ONTOLOGIES

The ontology is built entirely in Java using a set of JENA APIs, and RDF tools are used to describe the data. The Jena is a free and open-source Java framework for building Semantic Web applications. It provides extensive Java libraries for developing code that handles RDF, RDFS, RDFa, OWL, and SPARQL in line with published W3C recommendations. Jena includes a rule-based inference engine to perform reasoning based on OWL and RDFS ontologies. The PLE for content recommendation requires information about learner and learning materials. Appropriate user interfaces have been developed to collect this information from the learner and the instructor. When the learner starts using the LMS initially, the personal data is recorded using forms, and this information is fed into the "Learner" class of the ontology. A questionnaire consisting of 40 questions is given to the learner to identify the learning style, which is crucial in the developed ontology. A "LearnerLog" class is included in the ontology to track the learning path of each learner. The other part of the ontology combines the LO characteristics for the selected topics. The instructor finds out appropriate learning resources and annotates based on the IEEE LOM specification for each topic chosen. The instructors use appropriate user interfaces to feed this data into the "LearningMaterial" class of the ontology. The details about the developed ontology are described in the following subsections.

4.1 Sub-ontologies

The proposed ontology contains three sub-ontologies. Sub-ontology is a reusable module that is self-contained and logically consistent and tied to other sub-ontologies within the mother ontology. Here, the sub-ontologies are Learner, Learner Log, and Learning Material; each of them is explained subsequently.

4.1.1 Learner. The individual differences and preferences of learners should be considered to achieve personalization in learning environments. Here, we focused on both the static and dynamic characteristics of the learners for modeling them. The core concept in the developed ontology is a learner class in which each learner is represented with different object properties. The components of the learner class include personal information about the learner as well as the learning style of each learner. The data type properties of this class are student identification number, name, age, gender, branch, qualification, and background knowledge of the learners. Each learner has a learning style and the data type values of this attribute correspond to the four dimensions of FSLSM. These dimensions can be viewed as a continuum with one learning preference on the far left and the other on the far right. The FSLSM model classifies individuals in four dimensions as shown in Table 1.

The personal information together with the learning style constitutes the initial static profile for each learner. The graphical representation of the *Learner* sub-ontology is shown in Figure 3.

4.1.2 Learner Log. When a learner starts learning through the LMS, the learning path should be tracked, and this information is required for suitable content recommendation. A *Learners_Log* class is included in the ontology to keep the user login data. Each learner has a section in the

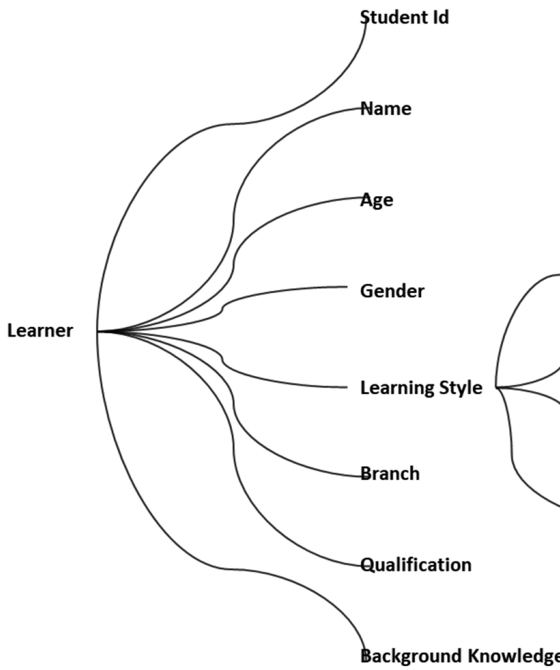

Fig. 3. Learner sub-ontology.

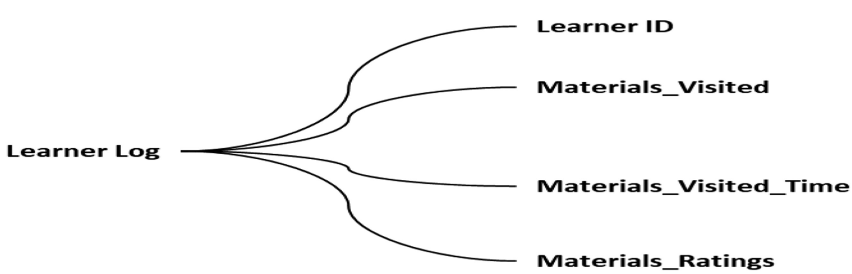

Fig. 4. LearnerLog sub-ontology.

Learners_Log class that is used for tracking the learning path of the learner in the LMS. The data type values of this class are materials_visited, materials_visited_time, and materials_ratings. This class's data attributes help to find the materials visited and the time spent by the learner on each learning material. The ratings given to the learning materials by the learners are also stored in this class. Since each learner has a section in the *Learners_Log* class, the *Learner* class is related to *Learners_Log* class with the relationship "hasSection." The *Learners_Log* class contributes to the dynamic part of the learner profile. The graphical representation of the LearnerLog sub-ontology is shown in Figure 4.

4.1.3 Learning Material. Resources metadata are fundamental for searching and recommending content in e-learning systems. Resources can be described in several ways. For instance, they can be described through standardized structured metadata or unstructured metadata, such as tags. Standardized metadata, such as IEEE LOM [78], specifies the metadata syntax and semantics of LOs. Due to its relatively wide acceptance in the academic environment and its extensive usage

Table 2. IEEE LOM Characteristics (Educational Category)

Data properties	Description
Identifier	A unique label that identifies this learning object in the LMS.
Title	A name was given to the learning object.
Structure	The organizational structure of the learning object. The possible values are, Atomic, Collection, Networked, Hierarchical, and Linear.
Format	Technical type of the learning object, which is useful to identify the software needed to use the learning object.
Interactivity Type	Mode of learning supported by this LO. Modes can be active, expositive, or mixed.
Learning Resource Type	Kind of the LO. Value of this element is an ordered set of types such as exercise, graph, simulation, questionnaire, index, diagram, figure, slide, table, narrative text, exam, experiment, problem statement, self-assessment, and lecture.
Interactivity Level	The degree to which learner can affect the behavior of the LO. The measure can be very low, low, medium, high, and very high.
Difficulty	The degree of hardness to work with the LO. The scale can be very easy, easy, medium, difficult, and very difficult.

by institutional repositories, we opted for IEEE LOM. The standard proposes 80 data elements grouped into nine categories: general, lifecycle, meta-meta data, technical, educational, rights, relation, annotation, and classification. It is essential to highlight that only the educational category of IEEE LOM is considered for modeling the LOs in our work. The educational category of LOM is enough to recommend suitable learning material to a learner based on the learning style. In addition to these fields, general information of a LO can be represented using the general category fields. The LOM characteristics that come under the educational category are consolidated in Table 2.

The "LearningMaterial" is a sub-ontology used in the developed ontology to represent LO metadata. Each LO has eight attributes that are recorded in the *LOM_Characteristics* class. In addition to this, the ontology contains a data type property to indicate the topics included in the LMS. The graphical representation of the *LearningMaterial* sub-ontology is shown in Figure 5.

Ontology, together with a set of individual instances of classes, constitutes a knowledge base. Therefore, defining appropriate classes, objects, and data properties are significant while constructing ontologies. The class, object, and data properties of the developed ontology are consolidated in Table 3.

In this article, we are trying to address the pure cold-start problem in content recommenders using ontologies. The ontology has been built with all the necessary attributes required for the content recommendation in PLEs. Appropriate learner groups have to be extracted from the ontologies and fed to the OB recommender system to make relevant recommendations in the pure cold-start condition. Along with this, semantic mapping rules are built between learners and learning objects to fine-tune the cold-start recommendations. The semantic processes involved to generate the top N recommendation are explained subsequently.

5 SEMANTIC PROCESSES

Semantic processes are widely used for knowledge representation in the form of a relationship between data objects. In this study, we have utilized different semantic processes to get precise data essential for the recommendation process.

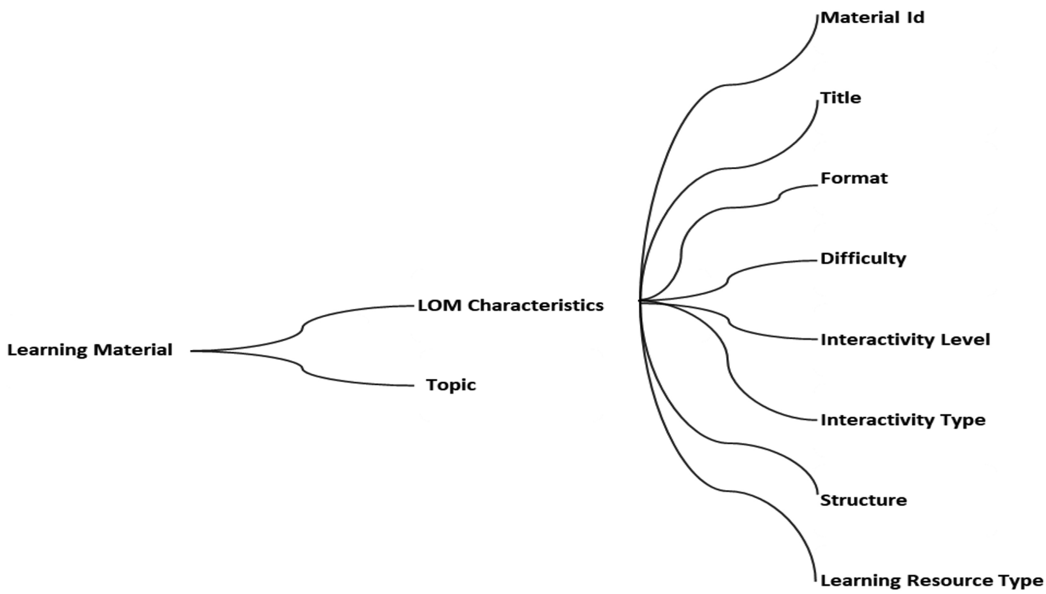

Fig. 5. Learning material sub-ontology.

Table 3. Class, Object, and Data Properties of the Developed Ontology

Class	Object	Data
Learner	has Character	Student Id, Name, Age, Gender, Branch, Qualification, Background Knowledge.
Learning Style	has Learning Style	Active/Reflective, Sensitive/Intuitive, Visual/Verbal, Global/Sequential.
Learner Log	has section	Materials_visited, Material_visited_time, Material_ratings.
LOM Characteristics	Visited Material	Material Id, Title, Format, Difficulty, Interactivity Level, Interactivity Type, Structure, Learning Resource Type.

5.1 Computing Similar Learner Groups

Adopting ontologies as the basis of the learner profile is vital in addressing the cold-start problem in PLEs. It allows the initial learner characteristics to be matched with prior knowledge in the ontologies and relationships between them. The advantage of creating domain ontology is its ability to describe a semantic model of the data combined with the related domain knowledge. The most recognized semantic query for information retrieval from ontologies is SPARQL [79]. A query has to be defined to run each of the semantic processes in the system. For executing each SPARQL query against the ontology and visualizing the results, a Jupiter notebook with live Python code was used in our study. The Jupyter Notebook is an open-source web application that allows creating and sharing documents that contain live code, equations, visualizations, and narrative text.

An example of a SPARQL query created with this application to extract similar learners is given below.

```
PREFIX onto: <http://elear2.org/onto.owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT * WHERE {?name onto:Qualification ?Qualification.
?name onto:Background_Knowledge ?Background_Knowledge.
?name onto:Global_Sequential ?Global_Sequential.
?name onto:Visual_Verbal ?Visual_Verbal.
?name onto:Active_Reflective ?Active_Reflective.
?name onto:Sensitive_Intutive ?Sensitive_Intutive.
?name onto:Age ?Age.
?name onto:Name ?Name.
FILTER ((?Active_Reflective ="1"^^xsd:integer)
        && (?Sensitive_Intutive ="1"^^xsd:integer)
        && (?Visual_Verbal ="1"^^xsd:integer)
        && (?Global_Sequential ="0"^^xsd:integer)
        && (?Background_Knowledge ="0"^^xsd:integer)
        && (?Qualification ="1"^^xsd:integer))}
```

Here, the objective is to extract the most similar learner group from the ontology whenever a new learner comes and starts using the LMS. The first and vital step is to identify the key learner attributes contributing to building the learner profile. The learner ontology contains demographic data such as age, gender, and other personal data related to their learning, such as qualification, branch, background knowledge, and learning style. Since this study addresses the pure cold-start problem alone, only the learner parameters learning style, qualification, and knowledge level were considered for learner grouping. The existing literature states that these parameters have more significance in improving the recommendation accuracy in cold-start conditions [20, 39, 40]. The addition of demographic attributes in the recommendation process may improve the personalization of learner recommendations. These learner parameters are not considered in this study, since their significance is less in the learner grouping process in pure cold-start condition. Whenever a new learner starts using the LMS, the similar learner groups are extracted from the ontology using SPARQL queries. The conditions for extracting learner groups using the SPARQL queries with prominent learner attributes are consolidated in Table 4.

Before running the SPARQL queries, the data pre-processing has been done on the input data. The pre-processing mainly includes datatype conversion of the input learner parameters (qualification, background knowledge), identifying the dominant learning style dimension from the learning style parameter. Both the qualification and the background knowledge are stored as string values in the ontology. These attributes were converted into corresponding integer index values (Example of knowledge level: Basic-0, Intermediate-1 and Expert-2). In the case of the learning style parameter, four dimensions are available for this parameter. The learning style parameter indicates how a learner prefers to learn, and it has four learning dimensions. The values for each dimension is stored in the ontology as integer values such as **Active**/Reflective=72, **Sensitive**/Intuitive=65, **Visual**/Verbal=34, and **Global**/Sequential=26. These integer values indicate the learning style of a learner as follows: Active=72 and Reflective= 28 (100-72), Sensitive=65 and Intuitive=35 (100-65), Visual=34 and Verbal=66 (100-34), Global=26 and Sequential=74. The dominant variable for each dimension (value >50) is assigned with a value "1" and the other variable with a "0" value. For a learner with the above learning style values (72, 65, 34, 26), the dominant values are calculated as (1, 1, 0, 0). After completing the necessary data

Table 4. Conditions for Extracting Similar Learner Groups Using SPARQL Queries

Learner parameters for learner grouping	Conditions of learner grouping
Qualification, Background Knowledge, Learning Style	IF Qualification = "{HS|Graduate|post-graduate}" ∩ Background Knowledge = "{Basic|Intermediate|Expert}" ∩ Learning Style = "{Active/Reflective, Sensitive/Intuitive, Visual/verbal, Sequential/Global}"
Qualification, Learning Style	IF Qualification = "{HS|Graduate| post-graduate}" ∩ Learning Style = "{Active/Reflective, Sensitive/Intuitive, Visual/verbal, Sequential/Global}"
Background Knowledge, Learning Style	IF Background Knowledge = "{Basic|Intermediate|Expert}" ∩ Learning Style = "{Active/Reflective, Sensitive/Intuitive, Visual/verbal, Sequential/Global}"
Learning Styles	IF Learning Style = "{Active/Reflective, Sensitive/Intuitive, Visual/verbal, Sequential/Global}"

	Name	Background_Knowledge	AR_Value	Visual_Verbal	Global_Seq	Qualification
0	L2	Intermediate	1	0	0	0
1	L122	Intermediate	1	0	0	0
2	L265	Expert	1	0	0	0
3	L273	Basic	1	0	0	0
4	L276	Expert	1	0	0	0
5	L277	Basic	1	0	0	0

Fig. 6. Sample output of a SPARQL query ran using Python code.

pre-processing steps, we ran multiple SPARQL queries with these key attributes for finding the most similar user group with the target user. The screenshot of the sample output obtained while running a SPARQL query over learner ontology is shown in Figure 6.

Ontologies are capable of providing dynamicity in learner profiling. The values of the learning style attribute can be updated as the learner progresses through the LMS. The learning path (how the learner progresses through different learning materials) is the feedback mechanism to update an existing learner's learning style in the learner ontology. The dynamicity in learner profiling will lead to the generation of quality recommendations and highly rated learning materials for each learner in the LMS. Since existing learners' ratings are also an input to the recommendation engine in the pure cold-start condition, the ontology, which is part of the RS, indirectly generates meaningful recommendations in pure cold-start condition.

5.2 Generating Top N recommendation List

The proposed recommendation model generates the top N recommendation list of learning objects for the target learner in pure cold-start condition, based on semantic rule mapping between the LO and the learner profile and the ratings given to a learning object by similar learners. The key input to the recommendation engine is the ontological domain knowledge about the learner and learning objects. In the pure cold-start condition, the primary learner attributes such as background knowledge and learning style are mapped with the LO attributes. The background knowledge (attribute values: basic, intermediate, expert) of the learner is mapped with the LO attribute difficulty level (attribute values: very easy, easy, medium, difficult, very difficult). For example, a learner with a knowledge level basic is mapped with a LO with difficulty level very easy/easy. Similarly, a learner's learning style is mapped with the LO attributes structure, format interactivity type, interactivity level, and learning resource type. A LO subset for the requested topic is extracted from the learning material ontology based on the rule set.

Similarly, based on similar learners' historical ratings (extracted by SPARQL queries), the most relevant LOs are retrieved from the domain ontology. Let "L" denote the set of all learners L = $\{l_1, l_2 \ldots, l_m\}$, let "LO" denote the possible subset of learning objects LO = $\{lo_1, lo_2 \ldots,\ lo_n\}$ that can be recommended (LO contains only the learning objects included for a particular topic requested by learner l). Let "O" be the set of all ontological domain knowledge O = $\{o_1, o_2 \ldots,\ o_p\}$ about the learner and learning objects. The ratings given to a learning object by the learners are indicated as "r." The possible rating values are measured on a numerical scale from 1 to 5 (1- poor, 2-average, 3- good, 4- very good, 5-excellent). Let f be the recommendation function of L, LO, and O. The top N refers to the sets of learning objects recommended by the recommendation engine. The recommendation function can be expressed as:

$$\mathrm{f} : L \times LO \times O \rightarrow \mathrm{top\,N}.$$

Algorithm 1 directs how the top N recommendation list is computed for the target learner in pure cold-start condition using a collaborative filtering technique.

ALGORITHM 1: Generating top N recommendation list in pure cold-start condition

Input:
- Ontology domain knowledge about learner and learning objects
- O = {learner, learner log, learning Material}
- Rating values r
- r $\in \{1, 2, 3, 4, 5\}$

Output:
- Top N recommendation list of LOs

Method:
1. Create a learner group containing the most similar "N" learners to the target learner by running SPARQL queries against learner ontology.
2. Get the LO subset for the requested topic from the learning Material ontology.
3. For each learner in the learner group obtained in step 1, get the corresponding rating value "*r*" for the LO subset obtained in step 2.
4. Generate the top N recommendations for the target learner with the highest rated LOs by the similar learners.

The target learner's final recommendations are generated by combining the recommendation lists created by semantic rule mapping and the collaborative filtering technique. The next step is to evaluate the results obtained by the semantic processes applied in the ontology. The details of the experiments conducted are explained subsequently.

6 EXPERIMENTS AND RESULTS

The learner satisfaction is measured by the ratings given to the learning object by the participants to evaluate the effectiveness of the proposed recommendation model in pure cold-start condition. A multivariate clustering tool is utilized to evaluate the proposed model's ability to identify similar learner groups.

Table 5. Learner Parameters for Clustering

Number of learner records	480
Learner Parameters	Qualification- String value – Indicates the highest educational qualification 1. HS 2. Graduate 3. Post graduate
	Background Knowledge – String value – Indicates the level of knowledge 1. Basic 2. Intermediate 3. Expert
	Learning Style – Integer value ranges from 1 to 100 1. Active/Reflective 2. Sensitive/Intuitive 3. Visual/Verbal 4. Global/Sequential

6.1 Dataset Description

For the evaluation purpose, the learner data is collected through the LMS, which is obtained through the interaction of learners with the LMS. The dataset used in this study is a real-world dataset consisting of 480 students' data, 468 annotated learning objects from the subject Data Structures and Algorithms, and 5,600 learner ratings. The learner data is collected from the students of two state universities of Kerala, India: Cochin University of Science and Technology (CUSAT) and APJ Abdul Kalam Technological University (KTU). The learner dataset contains undergraduate and postgraduate students' data undergoing Computer Science and Information Technology courses in the two universities. The dataset was collected in four months duration. The learning objects were annotated by experts of CUSAT and uploaded through the instructor interface of the developed LMS.

A multivariate clustering technique is utilized to measure the accuracy of learner similarity computation in this study. The overview of the data attributes used in the clustering technique is described in Table 5.

6.2 Experimental Setup and Evaluation

In this subsection, the experimental evaluation conducted to measure the proposed ontology-based recommendation algorithm's effectiveness is discussed. For conducting e-learning recommender systems experiments, the availability of public datasets is very rare [80]. It is not easy to compare different studies' performance results in e-learning recommender systems with the required accuracy. Therefore, we have not compared our approach with existing cold-start methods. Here, the experiments' main focus is to evaluate the accuracy in learner similarity computation and learner satisfaction with cold-start recommendations. In this study, the following evaluations were done:

- Learner similarity with multivariate clustering.
- Learner satisfaction in terms of ratings.

Table 6. Multivariate Clustering Tool Parameters

Multivariate clustering tool (Parameters)	Values and Measures
Clustering algorithm	K-means
Analysis fields	Learning style, Background Knowledge, qualification (different combinations of analysis field values are chosen for experimentations)
Number of clusters(k)	16 (optimum value)
Distance measure	Euclidean distance
Cluster's center initialization method	Random

6.2.1 Learner Similarity with Multivariate Clustering. Multivariate analysis is a method to find patterns and relationships between several variables simultaneously [81]. When many variables contribute to a problem, this kind of analysis is beneficial. It helps us to predict how a change in one variable is affecting other variables. Here, a multivariate clustering tool has been used to find the natural clusters in learner data. The clustering tool utilizes the unsupervised machine learning technique to group similar users. In this study, we have used a multivariate clustering technique with a k-means algorithm. The k-means algorithm is relatively simple to implement and easily adapts to new examples [82]. Another reason for choosing k-means is its best prediction accuracy in RSs [83]. The performance of the k-means algorithm usually deteriorates when outliers exist in the input data and the number of clusters (k) chosen is not appropriate. However, here, the chance of outliers in the input dataset is nil, and the prior knowledge of the data helps in choosing an appropriate number of clusters. The algorithm looks for a solution where all the features within each cluster are as similar as possible and all the clusters themselves are as different as possible. Feature similarity is based on the set of attributes that are specified in the Analysis Fields and Number of Cluster fields of the k-means algorithm. Both these inputs should be chosen so it will distinguish the feature's similarities and differences within a cluster. The details of the values and measures used in the clustering tool are consolidated in Table 6.

The learner data available in the ontology is exported to an Excel file for further processing to measure learner similarity computation accuracy using the clustering technique. Data pre-processing has been done to convert the string attributes (background knowledge and qualification) to corresponding integer index values. For the learning style attribute, pre-processing is not needed here, since the attribute values (Active/Reflective- 72, Sensitive/Intuitive-65, Visual/Verbal-34, and Global/Sequential-26) are required to be directly fed to the clustering algorithm. The details about the inputs and outputs of the clustering algorithm are explained below.

The k-means clustering algorithm's effectiveness mainly depends on the attributes chosen for analysis fields and the number of clusters to be created. Here, the analysis fields chosen are different combinations of the key attributes qualification, background knowledge, and learning style. In the same manner, these attributes were also taken for the semantic process.

The number of clusters was identified using the Elbow method, a prevalent technique to identify the optimum number of clusters for a given dataset using a k-means algorithm. The idea is to run the k-means algorithm for a range of clusters k (let us say from 1 to 10), and for each value, the sum of squared distances from each point to its assigned center is calculated. Here, the aim is to minimize the Sum of Squared Errors (SSE). The line graph of SSE is plotted against each value of k. If the line graph looks like an "arm," then the "elbow" on the arm is the optimum value of k. Here, for the learner dataset, the line graph of SSE is plotted for a range of values k=1 to 50 and is shown in Figure 7. The optimum number of clusters chosen from the elbow method is 16. Since the

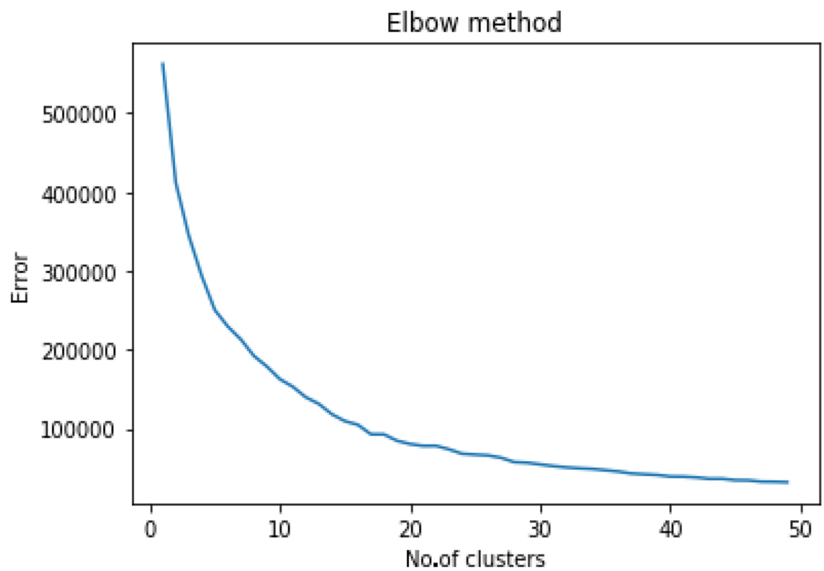

Fig. 7. Line graph (Elbow method).

dataset contains only 480 learners' data, we have executed the clustering algorithm with different k (number of clusters) values ranging from 10 to 20. There are slight variations in the result as the value of k changes. The best clustering accuracy is obtained for k=16, and this value is used for further experimentations.

Once the k value is fixed, the k-means clustering algorithm is iterated many times for different analysis field values combinations. The input attribute combinations are given in the clustering algorithm's analysis field, similar to that given in the SPARQL queries to retrieve similar learner groups. The different learner attribute combinations used for the experimentations are:

1. Background knowledge, Qualification, Learning style.
2. Background knowledge, Learning style.
3. Qualification, Learning style.
4. Learning style.

The multivariate clustering tool's default output is a new output feature class containing the fields used in the analysis. A new integer field called cluster_id identifies which cluster each feature belongs to. Here, the output (learner groups) obtained at each iteration of the clustering algorithm is compared with the corresponding learner group retrieved by SPARQL queries. The details of the results obtained are discussed under the "Experimental Results and Discussion" section. The three-dimensional projection of the output obtained while running the clustering algorithm with the learning style feature is shown in Figure 8.

The time taken by the clustering algorithm for learner similarity computation with varying cluster sizes is shown in Figure 9. From the figure, it is evident that the execution time varies with the learner parameters (Learning Style (LS), Background Knowledge (BK), and Qualification (QN)) used in the analysis field of the clustering algorithm. However, the execution time is not much varied with the number of clusters.

The time taken for computing the similar learner groups using the clustering technique and the SPARQL queries for different combinations of input learner parameters is consolidated in

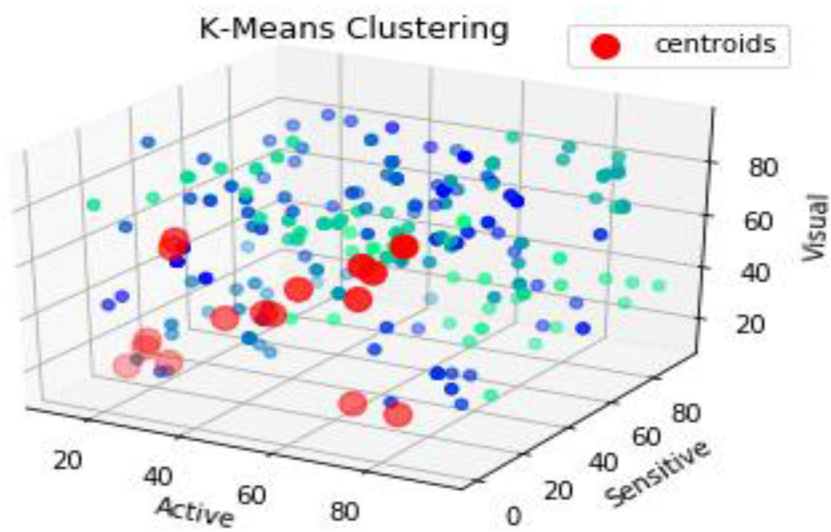

Fig. 8. Scatter plot (3-D) of clustering algorithm with the input parameter learning style.

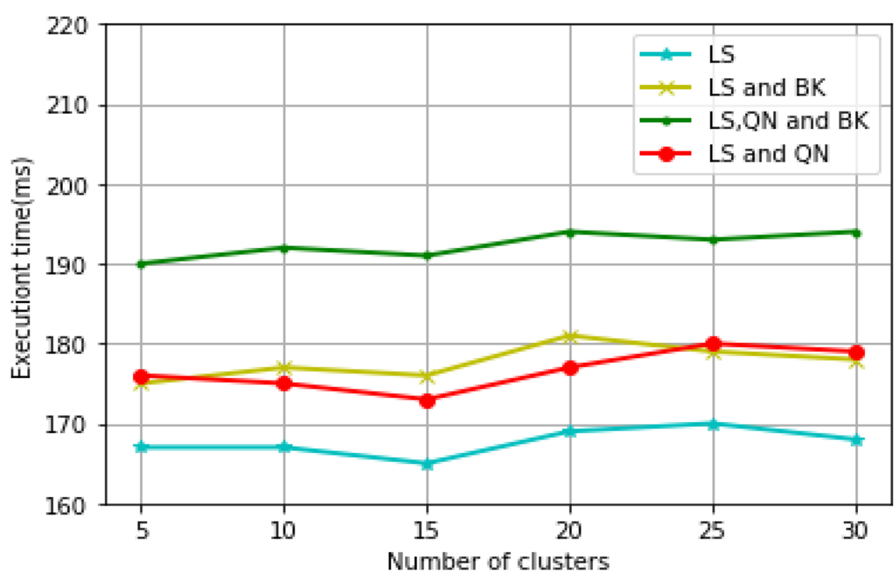

Fig. 9. Execution time of clustering algorithm (for different learner attributes) for varying cluster size.

Table 7. The execution time is much better in the case of SPARQL queries when compared with the clustering technique.

6.2.2 Learner Satisfaction in Terms of Ratings. The learner satisfaction with the proposed recommendation model is measured using the ratings given to the learning objects by the 40 participants of the experiments. The ratings given to a learning object by the learners are indicated as "r." The possible rating values are measured on a numerical scale from 1 to 5 (1- poor, 2-average, 3- good, 4- very good, 5-excellent). We have considered 100 ratings given to the LOs by 40 learners with the proposed recommendation approach for evaluating the learner satisfaction. Figure 10

Table 7. Execution Time for Learner Grouping Using Clustering and SPARQL Queries

Learner parameters	Execution time (Milliseconds)	
	Clustering(Average value)	SPARQL
Qualification, Background Knowledge, Learning Style	192.4	91.6
Qualification, Learning Style	176.7	84.9
Background Knowledge, Learning Style	178.3	85.2
Learning Styles	167.8	83.7

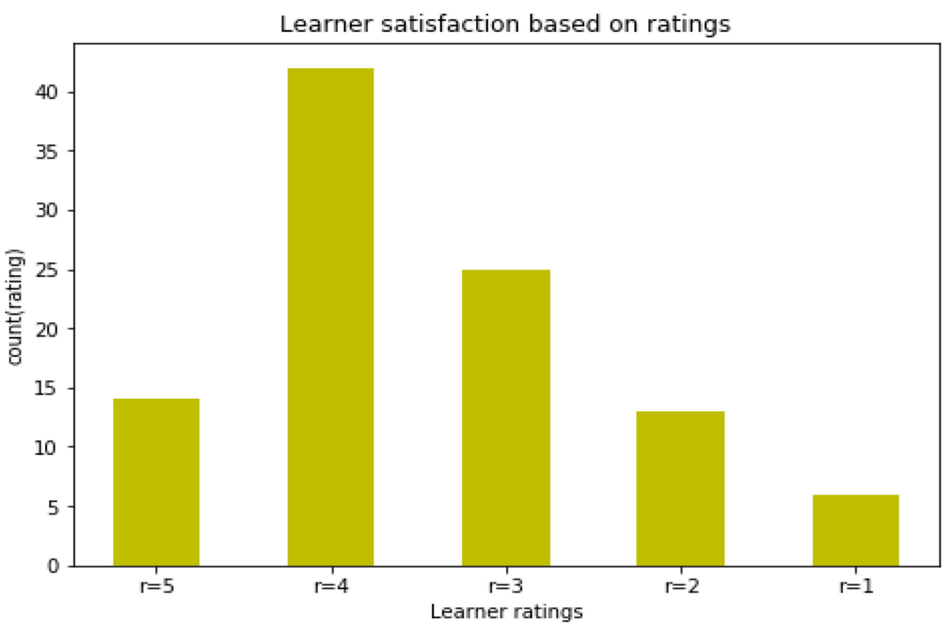

Fig. 10. Learner satisfaction based on ratings.

illustrates the satisfaction level of learners with the recommendations given by the ontology-based recommendation algorithm. In this study, learning objects rated 1 and 2 are considered irrelevant, while LOs rated three and above are considered relevant. From the bar plot, it is evident that 79% of learners (r=5(14), r=4(40), r=3(25)) are satisfied with the proposed recommendation model in the pure cold-start condition.

The experimental results obtained while running the algorithm for different learner parameter combinations are discussed in the coming subsection.

6.3 Experimental Results and Discussion

This section presents a detailed discussion of our experimental results of learner group estimation based on the learner parameters. In this work, we have grouped similar learners from the ontology, and results were compared using a multivariate clustering technique.

The key learner attributes included in the learner ontology are demographic data (age, gender), qualification, knowledge level, branch, and learning styles. Among these, we have taken only the prominent learner attributes qualification, knowledge level, and learning styles for learner grouping.

Table 8. Percentage of Relevant Records Retrieved from Ontology with Key Learner Parameters

Learner parameters	Relevant learner records (%)
Qualification, Background Knowledge, Learning Style	63.7
Qualification, Learning Style	67.6
Background Knowledge, Learning Style	76.4
Learning Styles	46.9

We have considered different combinations of learner parameters for extracting similar learners' groups from the ontology using multivariate k-means clustering. The learner groups extracted from the ontology using SPARQL queries are compared with the learner clusters formed by the clustering algorithm for each learner parameter combination. The percentage of the relevant learner records retrieved by the semantic model is calculated as in Equation (1).

$$\text{Percentage of relevant learner records} = \frac{\text{Number of relevant learners in the cluster}}{\text{Number of learners extracted from the ontology}} \quad (1)$$

For example, let the number of learners extracted from the ontology for a particular learner parameter combination is 21. If 14 out of these 21 learners are present in the corresponding learner cluster obtained using the clustering algorithm, then the percentage of relevant learner records=14/21=66.6%. The experimentation is repeated many times for each learner parameter combination, and the average value is taken as the final percentage of relevant records retrieved. The percentage of relevant records extracted from the ontology with different learner parameter combinations are given in Table 8. We got the maximum percentage of relevant learner records while combining the learner parameters background knowledge and learning style and the minimum percentage when using the learning style parameter alone.

We have extracted learner groups with different combinations of the key attributes. The qualification and the background knowledge contain three attribute values for each (Qualification: HS, Graduate, and Post Graduate; Background Knowledge: Basic, Intermediate, and Expert). For each learner, the FSLSM learning style attribute holds four integer values between 1 and 100 corresponding to the four learning style dimensions (Active/Reflective, Sensitive/Intuitive, Visual/Verbal, and Global/Sequential). For example, consider the learning style dimension Active/Reflective. If the Active value is 75, then the Reflective value of the learner should be $100-75 = 25$. Since the Active value is high, the learner is considered as an Active learner. When similar learner groups are extracted from the ontology using SPARQL queries, the information retrieved based on the learning style parameter is "whether a learner is Active or Reflective" (not the actual learning style dimension value). For an active learner, the learning style dimension value is chosen as "1" while for a reflective learner, the dimension value will be chosen as "0." However, in clustering, the actual integer values (between 1 and 100) of the learning style dimensions are inputted to the algorithm. The learning style attribute values are inputted in two different ways in the learner grouping techniques using clustering and SPARQL queries (clustering-integer (example: 75), sparql-binary (example: 1)). This difference in the representation of attribute value is a reason for a minor reduction in the learner grouping accuracy with SPARQL queries. The number of relevant records retrieved using SPARQL queries is less (46.9%) when considering learning style alone for learner grouping. Since learning style is a prominent learner attribute that affects the content recommendation process, it cannot be excluded from the key attribute list. However, better recommendations can be achieved by classifying the learners based on learning style (using range values for all the learning style dimensions) in the leaner sub-ontology. Thus, the accuracy of learner grouping can be further increased. The classification of learners based on learning styles will be included in our future line of work.

In this study, experimental calculations are made to evaluate the accuracy of learner groups' similarity computation. The proposed ontology-based RS overcomes the pure cold-start problem by combining the semantic mapping rules with a collaborative filtering technique. The learner similarity computation is vital in the collaborative filtering-based recommendation process. In most of the existing content RSs, machine learning techniques are utilized for learner grouping and predicting learner ratings. The experimental results show that the proposed model can build reliable learner groups from the ontology using SPARQL queries alone. The proposed recommendation framework's advantage is that it is simple and does not require data mining or machine learning techniques for learner grouping. The proposed RS achieved up to 76.4% accuracy in retrieving similar learner groups compared with the clustering technique. Moreover, the execution time taken for learner grouping using the SPARQL query is less than half the time compared to the clustering technique's execution time. Furthermore, the learner satisfaction achieved by the proposed model is commendable. 79% of the learners are satisfied with the proposed model's recommendations in pure cold-start condition. Since ontology is the base of the proposed model, it will generate more reliable and personalized recommendations by using ontological domain knowledge and learning path information of learners in the RS.

7 CONCLUSION AND FUTURE WORKS

Personalized learning environments need to consider learners' specific demands and requirements to achieve better learning outcomes. This article describes an ontology model that can be used to address the pure cold-start problem in content recommenders in a PLE. The developed ontology model conceptualizes both learner and learning object characteristics that are found to be relevant for the content recommendation in an adaptive learning environment. We considered the static and dynamic characteristics of a learner to model them. The ontological model that we have created supports personalization based on FSLSM. The learning objects were modeled using standardized IEEE LOM metadata. Further, we have created a semantic model with the developed ontology to group similar learners and map learner characteristics with LO characteristics. For building the semantic model, we have considered different learner parameters. We have found that learner parameters such as knowledge level and learning style are more prominent in finding similar learners' groups when new learners start using the LMS. For data evaluation purposes, a multivariate version of the k-means clustering algorithm has been used. The learner satisfaction with the proposed recommendation approach is measured using learner ratings. The experimental results show that the learners' group extracted by the semantic process matches that of the clustering output.

The model's accuracy may be further improved by categorizing the learners in the ontology model based on their learning style, which is proposed as our future work. Also, we plan to conduct the experiments with large learner datasets to find how the accuracy changes as the number of learners increases in the LMS. We plan to conduct experiments with more learner parameters such as age and gender to understand how these parameters affect the recommendation process. In this work, we have compared the results of learner similarity computation obtained using SPARQL queries with that of the k-means clustering technique. It is interesting to compare the results with other classification/clustering methods. Therefore, we are also planning to conduct experiments to compare the proposed model's effectiveness with other machine learning techniques.

REFERENCES

[1] E. T. Welsh, C. R. Wanberg, K. G. Brown, and M. J. Simmering. 2003. E-learning: emerging uses, empirical results and future directions. *Int. J. Train. Dev.* 7, 4 (2003), 245–258.

[2] D. Zhang, J. L. Zhao, L. Zhou, and J. F. Nunamaker Jr. 2004. Can e-learning replace classroom learning? *Commun. ACM* 47, 5 (2004), 75–79.

[3] K. H. Koh, E. Fouh, M. F. Farghally, H. Shahin, and C. A. Shaffer. 2018. Experience: Learner analytics data quality for an eTextbook system. *J. Data Inf. Qual.* 9, 2 (2018), 1–10.

[4] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez. 2013. Recommender systems survey. *Knowl.-based Syst.* 46 (2013), 109–132.

[5] L. H. Son. 2016. Dealing with the new user cold-start problem in recommender systems: A comparative review. *Inf. Syst.* 58, (2016), 87–104.

[6] L. Safoury and A. Salah. 2013. Exploiting user demographic attributes for solving cold-start problem in recommender system. *Lect. Notes Softw. Eng.* 1, 3 (2013), 303–307.

[7] Y. Koren and R. Bell. 2015. Advances in collaborative filtering. In *Recommender Systems Handbook*. Springer, Boston, MA, 77–118.

[8] X. Su and T. M. Khoshgoftaar. 2009. A survey of collaborative filtering techniques. *Adv. Artif. Intell.* 4 (2009), 19. Retrieved from https://downloads.hindawi.com/archive/2009/421425.pdf.

[9] M. J. Pazzani and D. Billsus. 2007. Content-based recommendation systems. In *The Adaptive Web*. Springer, Berlin, 325–341.

[10] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock. 2002. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 253–260.

[11] X. N. Lam, T. Vu, T. D. Le, and A. D. Duong. 2008. Addressing cold-start problem in recommendation systems. In *Proceedings of the 2nd International Conference on Ubiquitous Information Management and Communication*. 208–211.

[12] A. Hernando, J. Bobadilla, F. Ortega, and A. Gutiérrez. 2017. A probabilistic model for recommending to new cold-start non-registered users. *Inf. Sci.* 376 (2017), 216–232.

[13] A. Majumdar and A. Jain. 2017. Cold-start, warm-start and everything in between: An autoencoder based approach to recommendation. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN'17)*. IEEE, 3656–3663.

[14] A. Klašnja-Milićević, B. Vesin, M. Ivanović, and Z. Budimac. 2011. E-Learning personalization based on hybrid recommendation strategy and learning style identification. *Comput. Educ.* 56, 3 (2011), 885–899

[15] J. Buder and C. Schwind. 2012. Learning with personalized recommender systems: A psychological view. *Comput. Hum. Behav.* 28, 1 (2012), 207–216.

[16] F. Essalmi, L. J. B. Ayed, M. Jemni, and S. Graf. 2010. A fully personalization strategy of E-learning scenarios. *Comput. Hum. Behav.* 26, 4 (2010), 581–591

[17] N. S. Raj and V. G. Renumol. 2019. A rule-based approach for adaptive content recommendation in a personalized learning environment: An experimental analysis. In *Proceedings of the IEEE 10th International Conference on Technology for Education (T4E'19)*. 138–141.

[18] J. Y. Park, S. H. Joo, F. Cornillie, H. L. van der Maas, and W. Van den Noortgate. 2019. An explanatory item response theory method for alleviating the cold-start problem in adaptive learning environments. *Behav. Res. Meth.* 51, 2 (2019), 895–909.

[19] K. Pliakos, S. H. Joo, J. Y. Park, F. Cornillie, C. Vens, and W. Van den Noortgate. 2019. Integrating machine learning into item response theory for addressing the cold start problem in adaptive learning systems. *Comput. Educ.* 137 (2019), 91–103.

[20] J. K. Tarus, Z. Niu, and G. Mustafa. 2018. Knowledge-based recommendation: A review of ontology-based recommender systems for e-learning. *Artif. Intell. Rev.* 50, 1 (2018), 21–48.

[21] J. Joy, N. S. Raj, and V. G. Renumol. 2019. An ontology model for content recommendation in personalized learning environment. In *Proceedings of the 2nd International Conference on Data Science, E-Learning and Information Systems*. 1–6.

[22] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez. 2013. Recommender systems survey. *Knowl.-based Syst.* 46 (2013), 109–132.

[23] I. Barjasteh, R. Forsati, F. Masrour, A. H. Esfahanian, and H. Radha. 2015. Cold-start item and user recommendation with decoupled completion and transduction. In *Proceedings of the 9th ACM Conference on Recommender Systems*. 91–98.

[24] C. Wongchokprasitti, J. Peltonen, T. Ruotsalo, P. Bandyopadhyay, G. Jacucci, and P. Brusilovsky. 2015. User model in a box: Cross-system user model transfer for resolving cold start problems. In *Proceedings of the International Conference on User Modeling, Adaptation, and Personalization*. Springer, Cham, 289–301.

[25] S. T. Park, D. Pennock, O. Madani, N. Good, and D. DeCoste. 2006. Naïve filterbots for robust cold-start recommendations. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 699–705.

[26] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock. 2002. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 253–260.

[27] G. Shaw, Y. Xu, and S. Geva. 2010. Using association rules to solve the cold-start problem in recommender systems. In *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, Berlin, 340–347.

[28] P. Victor, M. De Cock, C. Cornelis, and A. M. Teredesai. 2008. Getting cold start users connected in a recommender system's trust network. In *Computational Intelligence in Decision and Control* 1 (2008), 877–882.

[29] Z. K. Zhang, C. Liu, Y. C. Zhang, and T. Zhou. 2010. Solving the cold-start problem in recommender systems with social tags. *Europhys. Lett.* 92, 2 (2010), 28002.

[30] Z. Gantner, L. Drumond, C. Freudenthaler, S. Rendle, and L. Schmidt-Thieme. 2010. Learning attribute-to-feature mappings for cold-start recommendations. In *Proceedings of the IEEE International Conference on Data Mining*. 176–185.

[31] J. Bobadilla, F. Ortega, A. Hernando, and J. Bernal. 2012. A collaborative filtering approach to mitigate the new user cold start problem. *Knowl.-based Syst.* 26, (2012) 225–238.

[32] H. J. Ahn. 2008. A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem. *Inf. Sci.* 178, 1 (2008), 37–51.

[33] A. L. V. Pereira and E. R. Hruschka. 2015. Simultaneous co-clustering and learning to address the cold start problem in recommender systems. *Knowl.-based Syst.* 82 (2015), 11–19.

[34] N. Silva, D. Carvalho, A. C. Pereira, F. Mourão, and L. Rocha. 2019. The pure cold-start problem: A deep study about how to conquer first-time users in recommendations domains. *Inf. Syst.* 80 (2019), 1–12.

[35] B. Chikhaoui, M. Chiazzaro, and S. Wang. 2011. An improved hybrid recommender system by combining predictions. In *Proceedings of the IEEE Workshops of International Conference on Advanced Information Networking and Applications*. IEEE, 644–649.

[36] A. N. Rosli, T. You, I. Ha, K. Y. Chung, and G. S. Jo. 2015. Alleviating the cold-start problem by incorporating movies Facebook pages. *Cluster Comput.* 18, 1 (2015), 187–197.

[37] M. R. Bouadjenek, H. Hacid, and M. Bouzeghoub. 2016. Social networks and information retrieval, how are they converging? A survey, a taxonomy and an analysis of social information retrieval approaches and and platforms. *Inf. Syst.* 56, (2016) 1–18.

[38] M. N. Moreno, S. Segrera, V. F. López, M. D. Muñoz, and Á. L. Sánchez. 2011. Mining semantic data for solving first-rater and cold-start problems in recommender systems. In *Proceedings of the 15th Symposium on International Database Engineering & Applications*. 256–257.

[39] H. Sobhanam and A. K. Mariappan. 2013. Addressing cold start problem in recommender systems using association rules and clustering technique. In *Proceedings of the International Conference on Computer Communication and Informatics*. IEEE, 1–5

[40] J. K. Tarus, Z. Niu, and A. Yousif. 2017. A hybrid knowledge-based recommender system for e-learning based on ontology and sequential pattern mining. *Fut. Gen. Comput. Syst.* 72 (2017), 37–48.

[41] C. He, D. Parra, and K. Verbert. 2016. Interactive recommender systems: A survey of the state of the art and future research challenges and opportunities. *Exp. Syst. Applic.* 56 (2016), 9–27.

[42] K. Munir and M. S. Anjum. 2018. The use of ontologies for effective knowledge modelling and information retrieval. *Appl. Comput. Inform.* 14, 2 (2018), 116–126.

[43] S. Geisler, C. Quix, S. Weber, and M. Jarke. 2016. Ontology-based data quality management for data streams. *J. Data Inf. Qual.* 7, 4 (2016), 1–34.

[44] G. Adomavicius and A. Tuzhilin. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.* 6 (2005), 734–749.

[45] X. Zhao, Z. Niu, W. Chen, C. Shi, K. Niu, and D. Liu. 2015. A hybrid approach of topic model and matrix factorization based on two-step recommendation framework. *J. Intell. Inf. Syst.* 44, 3 (2015), 335–353.

[46] S. Ouf, M. A. Ellatif, S. E. Salama, and Y. Helmy. 2017. A proposed paradigm for smart learning environment based on semantic web. *Comput. Hum. Behav.* 72 (2017), 796–818.

[47] N. Pukkhem. 2013. Ontology-based semantic approach for learning object recommendation. *Int. J. Inf. Technol.* 3, 4 (2013), 12.

[48] B. Bouihi and M. Bahaj. 2017. An ontology-based architecture for context recommendation system in E-learning and mobile-learning applications. In *Proceedings of the International Conference on Electrical and Information Technologies (ICEIT'17)*. IEEE, 1–6.

[49] B. Saleena and S. K. Srivatsa. 2015. Using concept similarity in cross ontology for adaptive e-Learning systems. *J. King Saud Univ.-Comput. Inf. Sci.* 27, 1 (2015), 1–12.

[50] M. B. Medland. 2007. Tools for knowledge analysis, synthesis, and sharing. *J. Sci. Educ. Technol.* 16, 2 (2007), 119–153.

[51] S. Fraihat and Q. Shambour. 2015. A framework of semantic recommender system for e-learning. *J. Softw.* 10, 3 (2015), 317–330.

[52] M. Harrathi, N. Touzani, and R. Braham. 2017. A hybrid knowledge-based approach for recommending massive learning activities. In *Proceedings of the IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA'17)*. 49–54.

[53] C. Obeid, I. Lahoud, H. El Khoury, and P. A. Champin. 2018. Ontology-based recommender system in higher education. In *Proceedings of the World Wide Web Conference*. 1031–1034.

[54] S. Shishehchi, S. Y. Banihashem, N. A. M. Zin, S. A. M. Noah, and K. Malaysia. 2012. Ontological approach in knowledge based recommender system to develop the quality of e-learning system. *Austral. J. Basic Appl. Sci.* 6, 2 (2012), 115–123.

[55] G. George and A. M. Lal. 2019. Review of ontology-based recommender systems in e-learning. *Comput. Educ.* 142 (2019), 103642.

[56] C. Wei, C. Huang, and H. Tan. 2009. A personalized model for ontology-driven user profiles mining. In *Proceedings of the International Symposium on Intelligent Ubiquitous Computing and Education*. IEEE, 484–487.

[57] L. Han and G. Chen. 2009. A fuzzy clustering method of construction of ontology-based user profiles. *Adv. Eng. Softw.* 40, 7 (2009), 535–540.

[58] M. Reformat and S. K. Golmohammadi. 2009. Updating user profile using ontology-based semantic similarity. In *Proceedings of the IEEE International Conference on Fuzzy Systems*. IEEE, 20–24.

[59] T. Sheeba and S. H. Begum. 2013. An ontology approach to represent user profiles in e-learning. *Int. J. Eng. Res. Technol.* 2, 2 (2013).

[60] B. Ciloglugil and M. M. Inceoglu. 2018. A learner ontology based on learning style models for adaptive e-learning. In *Proceedings of the International Conference on Computational Science and Its Applications*. Springer, Cham, 199–212.

[61] B. Ciloglugil and M. M. Inceoglu. 2012. User modeling for adaptive e-learning systems. In *Proceedings of the International Conference on Computational Science and Its Applications*. Springer, Berlin, 550–561.

[62] H. M. Truong. 2016. Integrating learning styles and adaptive e-learning system: Current developments, problems and opportunities. *Comput. Hum. Behav.* 55, (2016) 1185–1193.

[63] T. Sheeba and R. Krishnan. 2016. An ontological framework of semantic learner profile in an e-learning system. In *Proceedings of the International Conference on Brain Inspired Cognitive Systems*. Springer, Cham, 284–297.

[64] T. P. Liang, Y. F. Yang, D. N. Chen, and Y. C. Ku. 2008. A semantic-expansion approach to personalized knowledge recommendation. *Decis. Supp. Syst.* 45, 3 (2008), 401–412.

[65] S. Nafea, L. Maglaras, F. Siewe, R. Smith, and H. Janicke. 2016. Personalized students' profile based on ontology and rule-based reasoning. *ICST Trans.* 3, 12 (2016).

[66] W. Chen, Z. Niu, X. Zhao, and Y. Li. 2014. A hybrid recommendation algorithm adapted in e-learning environments. *World Wide Web* 17, 2 (2014), 271–284.

[67] R. M. Felder and L. K. Silverman. 1988. Learning and teaching styles in engineering education. *Eng. Educ.* 78, 7 (1988), 674–681.

[68] P. Polsani. 2003. Use and abuse of reusable learning objects. *J. Dig. Inf.* 14, 10 (2012). Retrieved from https://journals.tdl.org/jodi/index.php/jodi/article/viewArticle/89/88.

[69] B. Ciloglugil and M. M. Inceoglu. 2016. Ontology usage in e-learning systems focusing on metadata modeling of learning objects. In *Proceedings of the International Conference on New Trends in Education (ICNTE'16)*. 80–96.

[70] M. Al-Yahya, R. George, and A. Alfaries. 2015. Ontologies in e-learning: review of the literature. *Int. J. Softw. Eng. Its Applic.* 9, 2 (2015), 67–84.

[71] P. Raju and V. Ahmed. 2012. Enabling technologies for developing next-generation learning object repository for construction. *Autom. Constr.* 22, (2012), 247–257.

[72] S. Sosnovsky, I. H. Hsiao, and P. Brusilovsky. 2012. Adaptation "in the wild": Ontology-based personalization of open-corpus learning material. In *Proceedings of the European Conference on Technology Enhanced Learning*. Springer, Berlin, 425–431.

[73] S. Cakula and M. Sedleniece. 2013. Development of a personalized e-learning model using methods of ontology. *Procedia Comput. Sci.* 26, 0 (2013), 113–120.

[74] N. Pukkhem. 2014. LORecommendNet: An ontology-based representation of learning object recommendation. In *Recent Advances in Information and Communication Technology*. Springer, Cham, 293–303.

[75] Q. Han, F. Gao, and H. Wang. 2010. Ontology-based learning object recommendation for cognitive considerations. In *Proceedings of the 8th World Congress on Intelligent Control and Automation*. IEEE, 2746–2750.

[76] S. Fraihat and Q. Shambour. 2015. A framework of semantic recommender system for e-learning. *J. Softw.* 10, 3 (2015), 317–330.
[77] C. Huang, L. Liu, Y. Tang, and L. Lu. 2011. Semantic web enabled personalized recommendation for learning paths and experiences. In *Proceedings of the International Conference on Information and Management Engineering*. Springer, Berlin, 258–267.
[78] U. Risk. 2002. Draft standard for learning object metadata. *IEEE Standard* (2002), 1484, 1. Retrieved from http://www.dia.uniroma3.it/~sciarro/e-learning/LOM_1484_12_1_v1_Final_Draft.pdf.
[79] E. Prud'hommeaux and A. Seaborne. 2006. SPARQL query language for RDF. w3c working draft. World Wide Web Consortium. http://www.w3.org/TR/rdf-sparql-query.
[80] N. Manouselis, H. Drachsler, R. Vuorikari, H. Hummel, and R. Koper. 2011. Recommender systems in technology enhanced learning. In *Recommender Systems Handbook*. Springer, Boston, MA, 387–415.
[81] C. Chatfield. 2018. *Introduction to Multivariate Analysis.* Routledge.
[82] J. Wu. 2012. Cluster analysis and K-means clustering: An introduction. In *Advances in K-means Clustering*. Springer, Berlin, 1–16.
[83] S. Zahra, M. A. Ghazanfar, A. Khalid, M. A. Azam, U. Naeem, and A. Prugel-Bennett. 2015. Novel centroid selection approaches for KMeans-clustering based recommender systems. *Inf. Sci.* 320 (2015), 156–189.

Received February 2020; revised October 2020; accepted October 2020