Intelligent Automation & Soft Computing
DOI: 10.32604/iasc.2023.031329
Article

Tech Science Press

Context-Aware Practice Problem Recommendation Using Learners' Skill Level Navigation Patterns

P. N. Ramesh[1,*] and S. Kannimuthu[2]

[1]Computer Science and Engineering, Karpagam Institute of Technology, Coimbatore, 641105, India
[2]Information Technology, Karpagam College of Engineering, Coimbatore, 641032, India
*Corresponding Author: P. N. Ramesh. Email: pnr.cse@karpagamtech.ac.in

Received: 14 April 2022; Accepted: 13 June 2022

Abstract: The use of programming online judges (POJs) has risen dramatically in recent years, owing to the fact that the auto-evaluation of codes during practice motivates students to learn programming. Since POJs have greater number of programming problems in their repository, learners experience information overload. Recommender systems are a common solution to information overload. Current recommender systems used in e-learning platforms are inadequate for POJ since recommendations should consider learners' current context, like learning goals and current skill level (topic knowledge and difficulty level). To overcome the issue, we propose a context-aware practice problem recommender system based on learners' skill level navigation patterns. Our system initially performs skill level navigation pattern mining to discover frequent skill level navigations in the POJ and to find learners' learning goals. Collaborative filtering (CF) and content-based filtering approaches are employed to recommend problems in the current and next skill levels based on frequent skill level navigation patterns. The sequence similarity measure is used to find the top k neighbors based on the sequence of problems solved by the learners. The experiment results based on the real-world POJ dataset show that our approach considering the learners' current skill level and learning goals outperforms the other approaches in practice problem recommender systems.

Keywords: Recommender systems; skill level navigation pattern; programming online judge; collaborative filtering; content-based filtering

1 Introduction

Automation is currently gaining ground in a variety of fields like Digital Assistance [1], Information Security [2–4], Medical [5,6], Psychology Management [7], Smart Cities [8], education and training [9]. One of the best tools for learning to program is programming online judge (POJ), which allows students to practice programming and logic building. POJ is an automation tool that checks the code for particular problem's correctness. Educational institutions and industries use POJ to assess students' computer programming and problem-solving abilities. Several research studies have been proposed for the design, development, and improvement of POJ. A distributed and open-access online judge for Jinan University

(OOJJ) integrated with curriculum management and contest management systems to teach programming courses has been developed. The system is designed with a service-oriented architecture design pattern, which can be accessed by different kinds of clients quickly. The developed platform provides information about code plagiarism, number of courses, number of problems, runtime status, and many more to learners and teachers [10]. In research work, researchers developed a distributed learning assistant for data structures by integrating concepts of the Association of Computing Machinery-International Collegiate Programming Contest (ACM-ICPC) Online Judge (OJ) system and distributed systems. The system is most capable of managing courses, codes, and cheat detection [11]. An approach has been proposed to improve the existing OJ system. The system added new values like code quality-based grading, personalized feedback, and code plagiarism checking to the existing system [12]. An OJ for practicing MYSQL queries was developed to assist instructors of Database Management System courses in evaluating Structured Query Language (SQL) queries. The OJ MYSQL database is built using the Node.js and React.js frameworks, which enable comparisons between results of student queries and results of teacher-written key queries [13]. Several research studies and surveys have also focused on the impact of learning programming through POJ platforms. The influence of gamification aspects used in the OJ was investigated by authors of this work. The findings demonstrate that gamification in OJ creates a competitive and collaborative culture, which increases problem-solving skills in both fast and slow learners [14]. The results of a study show that learner ability and enthusiasm have improved in solving problems using the C language while using the OJ Platform [15]. Two research studies have also shown that practicing programming through POJ platforms improves learners' performance and problem-solving abilities more than existing learning approaches [16,17]. Since POJ platforms play a pivotal role in learning programming, their use has boomed and offer many benefits. On the other hand, the number of problems on POJ platform is growing rapidly, resulting in information overload. Users find it difficult to solve problems since they cannot choose the right problem that suits their interests and ability. Information overload causes learners to lose interest in learning and discourages them from using POJ portal to practice programming. The worldwide solution to information overload is to use a recommender system. Recommender systems suggest items that are suitable for the target user. Several research papers have proposed various approaches to recommender systems in e-learning. Generally, four methods are popular for recommending learning objects that best fit learners. The content-based filtering approach suggests using items based on target learner's previous preferences [18,19]. The Collaborative Filtering (CF) approach recommends items based on similar learner interests [20,21]. Sequential pattern-based recommender approaches recommend items based on Sequential Pattern Mining (SPM) [22,23]. Hybrid systems combine the benefits of the three approaches mentioned above [24,25].

The existing recommendation approaches in the field of e-learning are not suitable to recommend problems in POJ. Learner modeling, course modeling, and rating information are required in this field. Current POJ platforms don't have such information for modeling. On the other hand, in POJ platforms, the recommendation engine should consider the learning goals and current context of the target learner (difficulty level, topic currently practicing) to recommend problems. However, the existing approaches do not consider learning goals and current context to recommend objects. To understand the problem, consider a learner having interest in learning programming and the goal of cracking competitive programming contests of product companies. The learner is currently practicing problems in branching statements. The recommender system should consider the current topic, difficulty level, and learning path to move towards the learning goal. Even though the learner's goal is to learn data structures, the learner who is now learning and solving medium-level problems in branching statements are unable to solve problems in arrays. At the same time, solving problems related to object-oriented programming is not meaningful. In addition to the above two constraints, recommender system should consider difficulty level of the problem. Generally, the problems are tagged with topics (operators, branching, looping, arrays and more) and difficulty levels (school, basic, easy, medium, hard, and more). We use the term "skill level" to jointly represent both topic preference and difficulty level. Assume the problems are scattered in a 2D skill space where x-axis and y-axis are the topics and difficulty level, respectively. The recommender system should identify learner's current skill level and recommend problems from current and next skill levels in the suitable learning path for the learner's learning goals. The existing content-based e-learning recommender systems work based on the similarity between content and the quality of the content through learner ratings. CF approaches focus on similar learners and content liked by similar learners. Both the traditional approaches don't consider the learners' learning goal and current skill level. Based on the above scenario, target learners may face the following two issues:

i) The problem recommended to the target learner might be solved by other learners having same learning goals, but it may be very difficult to solve it with the learner's current skill level.

ii) The problem recommended to the target learner might be easy to solve, but it is irrelevant to the learner's learning goal.

The POJ system also lacks rating information since one out of every ten learners only give explicit ratings. Most recommender systems depend on implicit ratings. The implicit rating is derived from the frequency and duration of usage of the learning content in an e-learning platform. The most visited content is the most useful content. In POJ systems, because practice problems are different from learning content, it is hard to figure out an implicit rating for a problem. Learners often practice problems at a skill level until they feel comfortable with it, then move on to next skill level in their learning path. Sequence Pattern Mining (SPM) could be used to mine the most interesting patterns in these skill level navigations. These patterns are helpful in identifying learners who have similar goals, and they can be used to identify the next skill level towards learning goals of a target learner. To address the issue of lack of rating information, the number of attempts taken to solve a problem can be considered as an implicit rating of the learner of the problem. Using the number of attempts as an implicit rating makes sense because if it takes a learner more attempts than usual to solve a problem that means the problem isn't suitable for them right now.

This work proposes a novel approach named "context-aware practice problem recommendation using learners' skill level navigation patterns". The proposed work performs skill level navigation pattern mining to derive frequent skill level navigation patterns, which gives more insights into learning goals and learning paths. Top k similar learners are identified based on similarity of the sequence of problems solved by users, and implicit ratings are derived based on the number of attempts to solve problems.

The proposed work uses the above information to implement collaborative filtering and content-based approaches in suitable skill levels for effective recommendations. Finally, our approach is evaluated with the real-world POJ system's dataset to show the performance compared to other traditional approaches.

This paper is organized as follows: Section 2 focuses on necessary background details to understand our research in practice problem recommendation. Section 3 describes proposed approach to the recommendation of problems in POJ platforms. Section 4 shows the experimental setup and analysis of our proposed work. Section 5 gives the conclusion.

2 Background

This section presents previous research works that aid in understanding our work on practice problem recommendation in POJ platforms. Subsection 2.1 refers to e-learning recommendation scenarios and research works. Subsection 2.2 refers to the impact of sequence-based approaches employed in the recommendation of e-learning content. Subsection 2.2 refers to existing methods in problem recommender systems. Subsection 2.3 refers to the topic sequence mining, which is the inspiration for our skill navigation pattern mining-based practice problem recommendation in POJ.

2.1 Recommendation in E-Learning

The recommendations in e-learning system are used to recommend different learning objects like materials, books, videos, challenges to solve, web pages, topics, and courses. Various research works have focused on each specific learning object because each type of object recommendation requires different features to be considered. Material recommendation based on material similarity and good learner ratings were focused on in the work. The work creates a content profile and a rating profile that are fed into a hybrid recommendation engine [18]. Only best learners' ratings are used to make sure that the ratings are genuine, so only the best content is used. Web Content recommendation for Learners based on their ability, interest, and content readiness to prepare the material. Existing web search engines retrieve based on search keywords, which doesn't consider the learning behavior and competency of learners. The work aims to supplement Web search engines with personalized search result recommendations that are tailored to students' learning abilities and activities [26]. The authors have proposed a book recommendation system for e-learners. They have used explicit and implicit ratings in a collaborative-filtering algorithm which utilizes the quick sort algorithm to improve speed of the book recommendation system [27]. The problem recommender system suggests practice problems in Online Judge. The CF approaches have been widely used in problem recommendation systems which focus on the learners' interests based on the peer learners' interests. But no recommendation system has considered the current skill level of a learner to solve recommended problem [28–31]. Other than learning content, recommender systems are used to recommend topics for course design and also to recommend courses. The topic recommendation system helps in curriculum design [32,33]. Several research studies have been focused on course recommendations to students [34,35].

E-learning recommender systems are tasked with recommending learning objects to learners. The e-commerce recommendation system considers users' preferences, but it is inadequate for e-learning recommender systems. An e-learning recommendation system should consider learner knowledge level, performance, pedagogical preference, the topic of interest, learning path, and more based on the type of object it recommends. The research focused on content recommendation systems based on learner activities and performance. The system used both content and CF to avoid cold-start problems [36]. The personalized learning object recommendation system adapted to learners' dynamic preferences is designed based on self-organization behavior. Self-organization behavior modeling is a content-based approach that moves relevant learning objects of current context toward target learner [30]. The multimedia content recommendation system was developed based on Content Based Convolutional Neural Network (CBCNN) to recommend multimedia learning objects [37]. The system addresses the problem of extracting information from multimedia content. The learning path helps to scale the system by recommending learning objects from the internet, which has large number of scattered learning objects. A system that generates and recommends online learning paths was developed by researchers [38].

2.2 Sequence Pattern Based Recommendation in E-Learning

Various sequence mining techniques have been used in e-learning recommendation systems, demonstrating that sequential patterns have a positive impact on the accuracy of recommender systems. The majority of successful e-commerce recommender systems failed to recommend learning content in an e-learning system. The fundamental reason for failure is that those systems do not take into account learners' changing learning behavior. The recommender system, proposed as a combination of ontology and SPM, overcomes change in learners' learning behavior problems. The system uses the CF technique to generate top N suggestions after creating ontology to represent learning items and learners. Finally, SPM is utilized to generate final recommendations. The weighted SPM approach is used in the system to identify the importance of the sequence. This hybrid system outperformed other existing methods in e-learning content recommendation [39].

A SPM approach is introduced to get learners' different learning styles. These learning styles help to build a personalized e-learning system that suggests learning objects based on learner learning behavior. The algorithm proposed by the authors is employed to identify the different learning styles by collecting data from learners, categorizing learners' behavior based on the sequence of actions logged in the Massive Open Online Course portal, and applying SPM to identify frequent sequential patterns in all dimensions [40]. A customized recommender system is introduced to recommend learning materials based on the sequential patterns identified by the Apriori All algorithm. In the system, authors proposed a trust-based recommendation system for the e-learning platform Protus. The trust-based weighted mean value is used in the system to differentiate significance of learning sequence. This integrated sequence mining approach produced better performances than the existing systems [41].

2.3 Recommendation in POJ Tools

The extended matrix represents how a learner tries to solve a problem. Research works showed a better performance in recommendation of problems in an Online Judge portal. Still, learning paths and learning goals are not considered in the recommendation systems. Tab. 1 summarizes existing approaches for problem recommendation in POJ platform.

Table 1: Summary of existing approaches in problem recommendations and short comings

Research work	Objective	Short comings
[28]	A simple recommender system recommends only one problem in POJ. The system is designed using stored procedures in a database.	The work is suitable only for the programming contest and a basic system whose accuracy is not guaranteed.
[29]	A CF-based problem recommender system was proposed based on evaluation of other traditional approaches.	CF only considers the rating. Systems do not consider learning goal and current skill level of the target learner.
[30]	The three-layer CF approach works based on user-preference-item method.	
[31]	The extended matrix used in the work gives more insights into learner-problem interaction.	

2.4 Recommendation using Topic Sequence Mining

The primary goal of this research work is to analyze the impact of considering learning paths across different topics and difficulty levels of problems. The research inspires idea of working on the topic sequence mining in video recommendation and utilizing sequence signals in user behavior sequence to recommend items in an e-commerce portal [42,43]. Previously, the relationship between inter-topic navigation was discovered using unsupervised topic modeling and sequence mining, and it was used to generate scalable and accurate video recommendations in Educational Platform. The work used a content-based recommendation system and topic sequence frequent pattern mining. Candidates generated from content-based approach are re-ranked using inter-topic sequential patterns to recommend relevant and diversified videos to the user. The latest work uses transition of user behavior sequence mining to provide effective recommendations on an e-commerce platform.

3 Methodology

We propose a recommender system that considers learners' learning goals and skill levels based on skill level navigation patterns, sequence similarity, and semantic relationships between problems to provide effective problem recommendations.

3.1 Overview of Proposed Framework

In this framework, we combine skill level navigation pattern mining, sequence similarity-based neighbors, CF, and content-based filtering approaches to improve accuracy of POJ recommendation system. Skill-level navigation patterns help to identify learners' learning goals and next skill levels (topic of interest and difficulty level) where the learner needs to solve a problem. Nearest neighbors are identified based on similarities between sequences of problems solved. CF is applied to problems in learner's current and next skill levels to generate candidate problems to recommend. Finally, content-based approach derives the new problems which are more similar to the problems solved by user and adds them to recommendations. The workflow of the proposed framework is presented in Fig. 1. The proposed approach also helps to recommend specific problems to the new learners. Problems having more accuracy in root node and its child node are recommended to the new learners. The details of the proposed approach are explained in following sections.

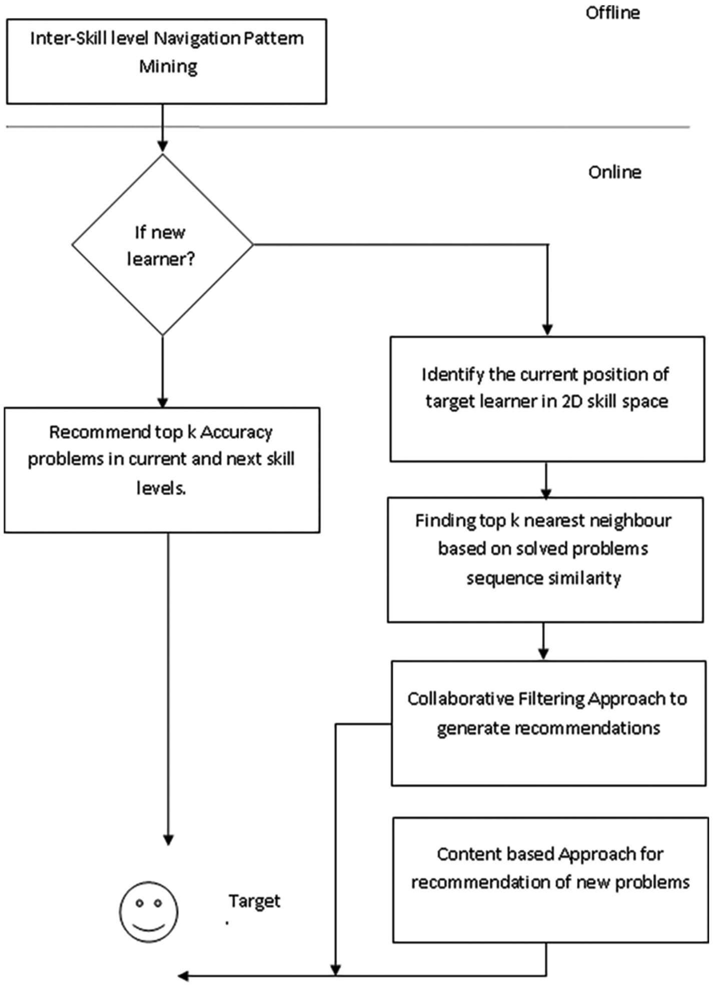
Offline
Inter-Skill level Navigation Pattem Mining
Online
If new learner?
Identify the current position of target learner in 2D skill space
Recommend top k Accuracy problems in current and next skill levels.
Finding top k nearest neighbour based on solved problems sequence similarity
Collaborative Filtering Approach to generate recommendations
Content based Approach for recommendation of new problems
Target

Figure 1: Overview of proposed research framework

The learner's current context (topic of interest and difficulty level) is more important in recommending practice problems in the POJ. In our proposed system, the term "skill-level" commonly represents both the topic and difficulty levels. To logically understand the scenario, we project the practice problems into a two-dimensional skill space where each column indicates topic of the problems and each row denotes difficulty level of problems. Fig. 2 demonstrates two-dimensional skill space. Each cell represents a skill level S_{ij} and contains p problems.

$$\text{Total Number of Problems} \quad P = \sum_{k=1}^{mXn} p_k \tag{1}$$

Hard	S_{1m}					S_{nm}
Mid Hard	S_{1m-1}					
. .						
Medium	S_{14}	S_{24} (Current Skill level)				
Easy	S_{13}					
School	S_{12}					
Beginner	S_{11}	S_{21}	S_{31}	S_{41}		S_{n1}
	IO	Control Structures	Array	STRINGS	. . .	NP Hard

Figure 2: Two-dimensional skill space

where

p_k indicates the number of problems in the cell k

k starts from S_{11} and ends at S_{nm}

Generally, learners navigate from one skill level to another (one cell to another) after practicing a target number of problems in each skill level. Common patterns of skill level navigation are successful learning paths derived from sequence pattern mining, and patterns are represented as graphs. Our proposed approach recommends problems from current skill level and next skill level based on the learners' navigation pattern. Our approach recommends problems from current skill level till target learner has an average number of problems solved by learners who successfully crossed at least next two skill levels. Later, it recommends problems from current and next skill levels. Common patterns of skill level navigation help recommendation system identify next skill levels where target learner needs to solve problems based on current skill level S_{ij}. The skill level navigation pattern mining also helps the system reduce memory while identifying similar learners for CF. Similar learners are identified based on sequence of solved problem from the root node to the current skill level node of the target learner.

3.2 Detail of Proposed Framework

3.2.1 Skill Level Navigation Pattern Mining

A sequential pattern mining discovers frequent sequences of skill level navigations or learning paths of the learner while learning programs and problem-solving in POJ. Hereafter, learning paths are referred to as "skill level navigation patterns". It has been proven that topic sequence mining helps scalable video recommendation systems [42]. Skill level navigation patterns are similar to topic sequential patterns, which describe how learners' interests move from one topic to another and from one difficulty level to another. Frequent skill level navigation helps the system predict learner's ability and interest.

Each problem P is projected in 2D skill space S_{ij}. Problem Ps is on topic i and difficulty level j. The problems solved by a user are presented in the submission history as follows.

$P_1(S_{11})$–$P_2(S_{11})$–$P_5(S_{11})$-$P_{20}(S_{11})$–$P_{32}(S_{12})$–$P_{35}(S_{12})$–$P_{36}(S_{12})$–$P_{42}(S_{21})$–$P_{45}(S_{21})$–$P_{46}(S_{13})$.

Our skill level navigation pattern mining derives interesting navigation patterns between the skill levels. For each user, skill level navigation is fetched from the submission history. Skill level navigation of the above submission history is S_{11}-S_{12}–S_{21}–S_{13}. These skill level navigations of all users are given as input to SPM algorithm, and output frequent sequence patterns are stored as a sequence pattern graph that can be further used by the system in implementing CF and Content Based algorithms.

However, skill level navigation pattern mining alone is not sufficient to predict problems. We use the steps below to clean noisy data before we use it to look for skill level navigation pattern patterns.

1. Consider only solved problems of learner's submission sequence.
2. Remove solved problems of skill level S_{ij} when problems of the same skill level S_{ij} were present earlier in the sequence submission.

Fig. 3 demonstrates skill level navigation pattern achieved through SPM. It tells us that learners who solve problems in skill levels <S_{11}, S_{21}, S_{12}> can complete problems in skill levels S_{22} and S_{13}. Skill navigation pattern mining creates a boundary in search space, which improves the accuracy of problem recommendations in a practice environment.

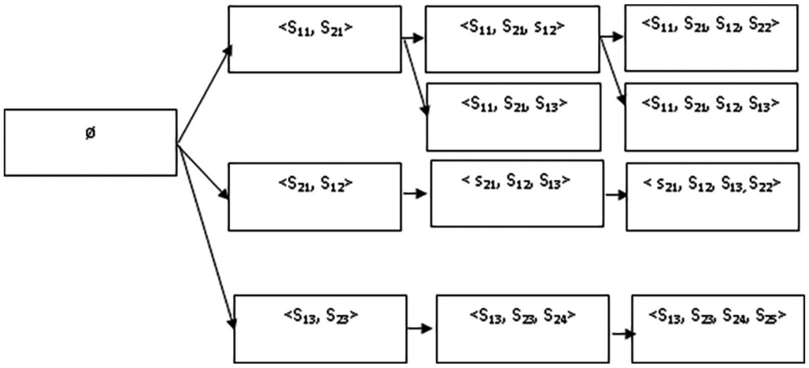

Figure 3: Skill navigation pattern

3.2.2 Categorization of Learners

New learners have just recently signed up, and they haven't completed enough problems in at least three skill levels. New learners are initially recommended with problems that have more accuracy in most of the starting skill levels of other learners. Once new learner solves enough problems at a skill level, the system recommends problems from next possible skill levels based on skill navigation pattern. Learner is not categorized as a new learner if they solve enough problems in at least three skill levels. For learners who are not categorized as new learners, we find current position in two-dimensional skill space based on problems solved by learner in the latest. We use window w which projects recent *l* problems solved by target learner, and choose top skill level at which learner has solved a greater number of problems.

See Fig. 4 which demonstrates that window w projects recent l problems solved by the target learner. The current skill level of target learner is identified as S_{31}.

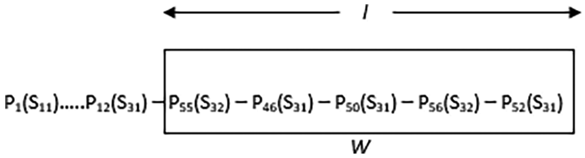

Figure 4: Window of recently solved problems

3.2.3 Top k Nearest Neighbour Based on Success Sequence Similarity

Based on the sequence of problems successfully performed, we apply top k-nearest neighbor approach to find similar learners of target learners. We assume that learners solving similar problems in a similar sequence have a similar ability to solve the problems. The problems to be solved will depend on problems solved in the past. A transition-based sequence matrix is used to represent sequence structure and a cosine similarity measure to calculate similarity between learners based on the sequence of the solved problems. Learners are at different skill levels; learners in the earlier skill level solved a smaller number of problems, and learners in the top skill level solved a greater number of problems. To balance the length of sequences, we consider only the problems solved by the user at their current skill level. Consider that the target learner is at skill level S_{24} and has solved problems in the sequence S_{11}, S_{12}, S_{21}, S_{22}, S_{23}, and S_{24}. Problems solved by the target learner ls_1 in skill levels S_{11}, S_{12}, S_{21}, S_{22}, S_{23}, and S_{24}, as well as problems solved by peer learner ls_2 prior to solving problems in skill level S_{24}, have been used to create a transition-based matrix. The solved problems belong to previous and current skill levels of the learner and are taken using skill level navigation of the learner identified in Section 3.2.1. If two learners' sequence is defined as ls_1 and ls_2, then similarity between the two learners is

$$Sim_{cos}(ls_1,\ ls_2) = \frac{\sum_{i,j} ls_1(i,\ j) ls_2(i,\ j)}{\sqrt{\sum_{i,j} [ls_1(i,\ j)]^2}\sqrt{\sum_{i,j} [ls_2(i,\ j)]^2}} \tag{2}$$

Several experiments have proven that transition-based representation of sequence data performs better in mining sequence data [44].

3.2.4 Collaborative Filtering Approach

We built recommendation system based on CF using a memory-based approach that calculates score for unsolved problems of target learner based on implicit rating taken from similar learners. We construct a p X q learner-problem interaction matrix M; p is the number of similar learners, including target learner, and q is the total number of problems in current and next skill level derived from skill level navigation patterns. To construct a interaction matrix M, we use functions f(i, j) and d(i, j), which calculate implicit rating by how learner i solved the problem j. Eq. (3) shows the procedure to derive an implicit rating matrix.

$$M[i,\ j] = f(i,\ j) + d(i,\ j) \tag{3}$$

The functions f(i, j) and d(i, j) return 0, if the learner i has not attempted the problem j to solve. For the rest of the scenarios, f(i, j) and d(i, j) return as follows:

$$f(i,\ j) = \begin{cases} 1, & if\ n \geq 1\ and\ not\ solved \\ 2, & if\ n \geq 2 * \alpha\ and\ solved \\ 3, & if\ n \geq 1.5 * \alpha\ and\ solved \end{cases} \tag{4}$$

$$f(i,\ j) = \begin{cases} 4, & if\ n \geq\ \alpha\ and\ solved \\ 5, & if\ n <\ \alpha\ and\ solved \end{cases} \tag{5}$$

$$\mathrm{d(i,\ j)} = \begin{cases} 1, & \textit{if } n \geq 1 \textit{ and not solved} \\ 2, & \textit{if } n \geq 2 * \beta \textit{ and solved} \\ 3, & \textit{if } n \geq 1.5 * \beta \textit{ and solved} \end{cases} \tag{6}$$

$$d(i,\ j) = \begin{cases} 4, & \textit{if } n \geq\ \beta \textit{ and solved} \\ 5, & \textit{if } n <\ \beta \textit{ and solved} \end{cases} \tag{7}$$

where

n is the number of attempts taken by learner i to solve problem j.

α is the average number of attempts taken by a learner i to solve problems.

β is the average number of attempts taken to solve a problem j by learners.

f(i, j) is a function that calculates rating based on how difficult the problem is when compared to other

problems solved by user i.

d(i, j) is a function that calculates a rating based on how the user is skilled to solve a problem when compared to other users who solved the problem j.

We calculate recommendation score $rscore_j$ for each problem j not solved by target learner u according to similar learners identified in Section 3.2.3. The recommendation score $rscore_j$ is calculated as the sum of product similarity between target learner u and neighbor learner who solved the problem j and implicit rating of j by neighbor learner. Eq. (8) shows the recommendation score calculation.

$$\mathrm{rscore_j} == \sum_{\mathrm{i=1}}^{\mathrm{k}} \mathrm{sim(u,\ i)} * \mathrm{M[i,\ j]} \tag{8}$$

Finally, top N problems with maximum scores are selected for recommendation to target learner u.

3.2.5 Content-based Approach for Recommendation of New Problems

In addition to problems from CF approach, we recommend that new problems in current and next skill levels of target learner by content-based filtering approach. In content-based approach, we consider attributes of problems like the topic, difficulty level, author, editor, and contest. Each problem is represented as a vector, and size of the vector is equal to the number of non-unique attribute values of all problems. Vector values are filled with Boolean values; if specific attribute value is present in the problem, one else is zero. We use a cosine similarity measure to calculate similarity between the two problems. In offline, for each problem, k similar new problems are stored along with their similarity values. Content-based filtering algorithm works as follows to recommend new problems in current and next skill levels to target learner: Solved Problems of the target learner and high implicit ratings (6 to 10) are considered as set P and a set of candidate problems C is built as union of k-most similar new problems for each problem $j \in P$, excluding the problems already in P. The next step is to compute similarity between each problem $c \in C$ and all problems in the P. The formula for computing similarity between c and all problems in P is

$$sim(c,\ P) = \sum_{j=1}^{n} \mathrm{sim(c,\ j)} \tag{9}$$

Finally, problems in *C* are sorted according to similarity value computed in the previous step. Problems with similarity values above threshold t are recommended, along with problems selected in CF approach to target learner u.

4 Experimental Analysis

4.1 Experimental Setup

An experiment was conducted using a real-world programming online judge (POJ) dataset consisting of 73 learners, 3925 problems, and 36235 submitted solutions to evaluate the proposed method. To implement skill level navigation pattern mining, find neighbors based on sequence similarity, and derive implicit ratings based on number of attempts as discussed in Section 3, we need complete history of submissions, which includes all the attempts of the learners. The dataset is crawled from the real-world POJ, which consists of complete submission history of its 73 users. The submission history includes all failed and successful submissions of users. Detail of program submission in history helps to get insights into the learner and problem characteristics. The problems in the dataset are tagged with 22 different topics and 6 different difficulty levels. Hence, problems in the dataset have been scattered across 132 skill levels. The mean value of each learner's number of problems solved is 140, with a standard deviation of 103. We implicitly derived rating for each problem interacted with learner through attempts taken to solve the problem. This implicit rating (1 to 10) shows how the problem is suitable to the learner's current context. Problems that received a rating of 10 are most suitable to the learner in the current context, while those that received a rating of 1 are highly unsuitable to learners in the current context. Submission history of 73 users has been taken and formatted as a suitable sequence to input SPM Algorithm. The offline module skill level navigation pattern mining is performed using the SPM Framework (SPMF: a Java open-source data mining library) with a maximum pattern length of 5. The navigation patterns over 132 skill levels are derived as output. Online modules for the experiment have been implemented using Python data science and machine learning libraries. The length l of the latest window w is set to 6 and based on the last 6 problems, current skill level of the target user is identified. The k value is set from 6 to 16 in neighbor identification process. Two common recommender system approaches, named CF and SPM, were also evaluated using the same dataset.

Collaborative Filtering Approach: CF-based approach is adopted using implicit rating matrix derived from the learner-problem interaction.

Sequential Pattern Mining Approach: SPM-based approach is adopted using sequence of solved problems derived from the submission history of users.

In the specified dataset, 80% is taken as a training set and remaining 20% is taken as a test set to assess performance of recommendation approaches. The above experiments showed that proposed approach to Sequence Enabled Practice Problem recommendation in POJ outperforms the CF-based and SPM-based recommendation approaches.

4.2 Evaluation Metrics

To evaluate accuracy of the proposed approach, we use metric F1 which is derived from precision and recall. Precision represents number problems are relevant among recommended problems, and recall represents number of relevant problems are recommended. If learner has solved the recommended problems, and the rating extracted is six or above, then it is considered relevant. Furthermore, if the learner has solved or not solved the recommended problem, and extracted rating is below six, then it is considered irrelevant.

$$F1 = \frac{2 * precision * recall}{precision + recall} \tag{10}$$

$$Precision = \frac{\sum_{i=1}^{ut} |TS(P_i) \cap R(P_i\,)|}{\sum_{i=1}^{ut} |R(P_i)|} \tag{11}$$

$$Recall \quad \frac{\sum_{i=1}^{ut} |TS(P_i) \cap R(P_i\,)|}{\sum_{i=1}^{ut} |TS(P_i)|} \tag{12}$$

where

ut is the number of learners in test set

TS(P_i) is the number of relevant problems solved by learner i in test set

R(P_i) is the number of recommended problems to learner i

Precision and recall are calculated based on experimental results using Eqs. (11) and (12). TS(Pi) is number of relevant problems solved by learner i, and R(Pi) is the set of recommended problems for learner i, which consists of both relevant and irrelevant problems. Both Precision and Recall reflect accuracy of recommender system through F1 measure.

4.3 Result Analysis

We have evaluated the proposed and existing approaches based on metric F1 measure derived from precision and recall. It has been the main thing to look at following ways to see how accurate the proposed method is:

1. We have tested proposed approach with various sizes of k nearest neighbors and various Top N numbers of recommendations.
2. We have tested our proposed and previous approaches with a k nearest and a top n number of recommendations.
3. We have tested proposed approach toward recommending new problems to address cold-start problem.

Initially, we tested proposed approach by varying the size of nearest neighbor k in the range of seven to twelve. Same testing is conducted with five different Top N values (6, 8, 10, 12, 14, and 16). Tab. 2 shows F1 measures obtained from the experiment. The results show that as the size of nearest neighbors increase, accuracy improves, and in most cases, accuracy is saturated at a size of 10. Therefore, we use k = 10 for next experiments, comparing proposed approach with the previous one.

Table 2: AccuracyF1 of proposed approach obtained for different size of k

Nearest neighbor (k) Size of N	7	8	9	10	11	12
6	0.48	0.50	0.52	0.53	0.53	0.54
8	0.47	0.48	0.48	0.51	0.51	0.51
10	0.43	0.45	0.45	0.46	0.47	0.47
12	0.42	0.43	0.44	0.46	0.46	0.46
14	0.39	0.42	0.42	0.44	0.44	0.44
16	0.34	0.34	0.37	0.38	0.39	0.39

Next, we experimented to evaluate proposed approach, CF approach, and the SPM approach using real-world dataset. Experiments are conducted by varying the number of recommendations from 6 to 12. Tab. 3 shows the experiment results, and it clearly shows that our proposed approach outperforms other familiar approaches, CF and SPM approaches.

Table 3: Comparison of F1 Accuracy between proposed and previous approaches

Approaches	SPM based recommendation			CF based recommendation			Proposed approach		
Size of N	P	r	F1	P	r	F1	p	r	F1
6	0.4	*0.22*	**0.28**	*0.42*	0.32	**0.36**	0.5	0.56	**0.53**
8	0.36	*0.23*	**0.28**	*0.39*	0.33	**0.36**	0.47	0.56	**0.51**
10	0.30	*0.25*	**0.27**	*0.30*	0.37	**0.33**	0.39	0.57	**0.46**
12	0.25	*0.25*	**0.25**	*0.24*	0.40	**0.30**	0.38	0.58	**0.46**
14	0.23	*0.27*	**0.25**	*0.21*	0.41	**0.28**	0.35	0.58	**0.44**
16	0.18	*0.28*	**0.22**	*0.17*	0.41	**0.24**	*0.28*	0.6	**0.38**

Finally, our evaluation focused on how proposed approach recommends new problems to address the cold-start problem. Both the familiar approaches, CF and SPM-based approaches need interactional data on items; therefore, both approaches couldn't handle new items (problems). To evaluate proposed approach, new problems at different skill levels are examined using a test data set. In this evaluation, we only consider recommendation of new problems to existing learner since the new learners are recommended with top accuracy problems in root skill levels. Tab. 4 shows 65 times new problems are recommended to 20 learners, and each learner gets 3.25 new problems on average. We evaluated accuracy of recommendations in dealing with new problems, and Tab. 4 shows that proposed method has an accuracy of 0.39 in recommending new problems.

Table 4: New Problem recommendations to existing learners

No of new problems	Total number of existing learners	No of skill levels	Total number of problems recommended	Average number of new problems recommended	F1-Measure for new problems
41	20	33	65	3.25	0.39

5 Conclusion and Future Work

The research work analyzed impact of considering efficient learning paths in recommending practice problems in POJ. The learning path is derived from skill level navigation patterns obtained from sequential pattern mining. These sequential patterns clearly show future skill levels where target learner can solve problems according to the current context. CF approach works only on these skill levels to recommend problems. Additionally, we have implemented sequence-based nearest neighbor method, which is most suitable in the POJ scenario to identify similar learners. The research work also covers solutions to cold start and data sparsity problems. To utilize advantages of each approach, we integrate content-based, CF, and SPM approaches in the right and appropriate direction. The proposed system outperforms other traditional approaches in terms of accuracy. In future, we will be working on heterogeneous learning object recommendations for POJ platform. This means that the system can recommend both personalized learning materials and personalized assignment problems so that people can learn how to program and practice programming at the same time.

Funding Statement: We received no specific funding for this study.

Conflicts of Interest: We declare that we have no conflicts of interest to report regarding the present study.

References

[1] S. Paliwal, V. Bharti and A. K. Mishra, "AI chatbots: Transforming the digital world," in V. Balas, R. Kumar, R. Srivastava (Eds.), in *Recent Trends and Advances in Artificial Intelligence and Internet of Things, Intelligent Systems Reference Library*, Switzerland: Cham, Springer, vol. 172, pp. 455–482, 2020.

[2] X. R. Zhang, W. F. Zhang, W. Sun, X. M. Sun and S. K. Jha, "A robust 3-D medical watermarking based on wavelet transform for data protection," *Computer Systems Science & Engineering*, vol. 41, no. 3, pp. 1043–1056, 2022.

[3] X. R. Zhang, X. Sun, X. M. Sun, W. Sun and S. K. Jha, "Robust reversible audio watermarking scheme for telemedicine and privacy protection," *Computers, Materials & Continua*, vol. 71, no. 2, pp. 3035–3050, 2022.

[4] G. Usha, S. Kannimuthu, P. D. Mahendiran, A. K. Shanker and D. Venugopal, "Static analysis method for detecting cross site scripting vulnerabilities," *International Journal of Information and Computer Security*, vol. 13, no. 1, pp. 32–47, 2020.

[5] S. Kannimuthu and A. Arunkumar, "Machine learning based approach for corona virus disease recovery prediction," *International Journal of Aquatic Science*, vol. 12, no. 3, pp. 1188–1199, 2021.

[6] S. Kannimuthu, K. S. Bhuvaneshwari, D. Bhanu, A. Vaishnavi and S. Ahalya, "Performance evaluation of machine learning algorithms for dengue disease prediction," *Journal of Computational and Theoretical Nano Science*, vol. 16, no. 12, pp. 5105–5110, 2019.

[7] V. Arya and A. K. Mishra, "Machine learning approaches to mental stress detection: A review," *Annals of Optimization Theory and Practice*, vol. 31, no. 4, pp. 55–67, 2021.

[8] S. Rani, R. K. Mishra, M. Usman, A. Kataria, P. Bhambri *et al.,* "Amalgamation of advanced technologies for sustainable development of smart city environment: A review," *IEEE Access*, vol. 9, pp. 150060–150087, 2021.

[9] S. Agarwal and A. K. Mishra, "Deploying block chain in education: Security, challenges, and solutions,*"* in *Proc. of 5th Int. Conf. on Information Systems and Computer Networks (ISCON)*, India, pp. 1–5, 2021.

[10] J. Wu, S. Chen and R. Yang, "Development and application of online judge system," in *Proc. of Int. Symp. on Information Technologies in Medicine and Education*, Hokodate, Hokkaido, Japan, pp. 83–86, 2012.

[11] X. Fang, Y. Du and N. Li, "A distributed online data structure learning assistant system based on ACM-ICPC mode," in *Proc. of 7th Int. Conf. on Information Technology in Medicine and Education (ITME)*, Huangshan, China, pp. 310–312, 2015.

[12] W. Zhou, Y. Pan, Y. Zhou and G. Sun, "The framework of a new online judge system for programming education," in *Proc. of ACM Turing Celebration Conf., Association for Computing Machinery*, Shanghai, China, pp. 9–14, 2018.

[13] D. Puspitasari, P. P. Arhandi, P. Y. Saputra, Y. W. Syaifudin, H. A. Himawan *et al.,* "Online judge MySQL for learning process of database practice course," in *Proc. of 8th Annual Int. Conf. on Science and Engineering, IOP Conf. Series: Materials Science & Engineering*, Aceh, Indonesia, vol. 523, 2019.

[14] F. D. Pereira, A. Toda, E. H. T. Oliveira, A. I. Cristea, S. Isotani *et al.,* "Can we use gamification to predict students' performance? A case study supported by an online judge," in *Proc. of Int. Conf. on Intelligent Tutoring Systems*, Athens, Greece, vol. 12149, pp. 259–269, 2020.

[15] H. Wu, Y. Liu, L. Qiu and Y. Liu, "Online judge system and its applications in C language teaching," in *Proc. of Int. Symp. on Educational Technology*, Beijing, China, pp. 57–60, 2016.

[16] G. P. Wang, S. Y. Chen, X. Yang and R. Feng, "OJPOT: Online judge & practice oriented teaching idea in programming courses," *European Journal of Engineering Education*, vol. 41, no. 3, pp. 304–319, 2016.

[17] S. Li, Y. Zhuo, S. Feng and C. Yi, "The application of course-oriented online judge in data structure," in *Proc. of IEEE Symp. on Robotics and Applications*, Malaysia, pp. 687–690, 2012.

[18] K. I. Ghauth and N. A. Abdullah, “Learning materials recommendation using good learners’ ratings and content-based filtering,” *Educational Technology Research and Development*, vol. 58, no. 6, pp. 711–727, 2010.

[19] N. Albatayneh, K. Ghauth and F. Chua, “Utilizing learners’ negative ratings in semantic content-based recommender system for e-learning forum,” *Journal of Educational Technology & Society*, vol. 21, no. 1, pp. 112–125, 2018.

[20] J. Bobadilla, F. Serradilla and A. Hernando, “Collaborative filtering adapted to recommender systems of e-learning,” *Knowledge-Based Systems*, vol. 22, no. 4, pp. 261–265, 2009.

[21] F. Loll and N. Pinkwart, “Using collaborative filtering algorithms as e-learning tools,” in *Proc. of 42nd Hawaii Int. Conf. on System Sciences*, NW Washington DC, United States, pp. 1–10, 2009.

[22] J. K. Tarus, Z. Niu and A. Yousif, “A hybrid knowledge-based recommender system for e-learning based on ontology and sequential pattern mining,” *Future Generation Computer Systems*, vol. 72, pp. 37–48, 2017.

[23] S. Muruganandam and N. Srinivasan, “Personalised e-learning system using learner profile ontology and sequential pattern mining-based recommendation,” *International Journal of Business Intelligence and Data Mining*, vol. 12, no. 1, pp. 78–93, 2017.

[24] S. Wan and Z. Niu, “Hybrid e-learning recommendation approach based on learners’ influence propagation,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 5, pp. 827–840, 2019.

[25] J. Jeevamol and V. G. Renumol, “An ontology-based hybrid e-learning content recommender system for alleviating the cold-start problem,” *Education and Information Technology*, vol. 26, no. 4, pp. 4993–5022, 2021.

[26] M. M. Rahman and N. A. Abdullah, “A personalized group-based recommendation approach for web search in e-learning,” *IEEE Access*, vol. 6, pp. 34166–34178, 2018.

[27] E. U. Okon, B. O. Eke and P. O. Asagba, “An improved online book recommender system using collaborative filtering algorithm,” *International Journal of Computer Applications*, vol. 179, no. 46, pp. 975–8887, 2018.

[28] L. C. De Paula, A. G. De Oliveira Fassbinder and E. F. Barbosa, “A recommendation system to support the student’s performance in programming contests,” in *Proc. of IEEE Frontiers in Education Conf.*, Madrid, Spain, pp. 1–8, 2014.

[29] R. Yu, R. Cai, X. Du, M. He, Z. Wang *et al.,* “The research of the recommendation algorithm in online learning,” *International Journal of Multimedia and Ubiquitous Engineering*, vol. 10, no. 4, pp. 71–80, 2015.

[30] X. Yu and W. Chen, “Research on three-layer collaborative filtering recommendation for online judge,” in *Proc. of 7th Int. Green and Sustainable Conf.*, Hangzhou, China, pp. 1–4, 2016.

[31] R. Yera and L. Martínez, “A recommendation approach for programming online judges supported by data preprocessing techniques,” *Applied Intelligence*, vol. 47, no. 2, pp. 277–290, 2017.

[32] M. Ballera, I. A. Lukandu and A. Radwan, “Personalizing e-learning curriculum using: Reversed roulette wheel selection algorithm,” in *Proc. of Int. Conf. on Education Technologies and Computers*, Lodz, Poland, pp. 91–97, 2014.

[33] E. García, C. Romero, S. Ventura and C. de. Castro, “An architecture for making recommendations to courseware authors using association rule mining and collaborative filtering,” *User Model User Adapt Interaction*, vol. 19, no. 1–2, pp. 99–132, 2009.

[34] K. Nishino, Y. Iribe, S. Mizuno, K. Aoki and Y. Fukumura, “The development of a course recommendation system for e-learning students,” *International Journal of Knowledge and Web Intelligence*, vol. 3, no. 1, pp. 19–32, 2012.

[35] C. Vialardi, J. Bravo, L. Shafti and A. Ortigosa, “Recommendation in higher education using data mining techniques,” in *Proc. of Int. Conf. on Educational Data Mining*, Cordoba, Spain, pp. 191–199, 2009.

[36] D. Herath and L. Jayarathne, “Intelligent recommendations for e-learning personalization based on learner’s learning activities and performances,” *International Journal of Computer Science and Software Engineering*, vol. 7, no. 6, pp. 130–137, 2018.

[37] J. Shu, X. Shen, H. Liu, H. B. Yi and Z. Zhang, “A Content-based recommendation algorithm for learning resources,” *Multimedia Systems*, vol. 24, no. 2, pp. 163–173, 2018.

[38] D. Shi, T. Wang, H. Xing and H. Xu, “A learning path recommendation model based on a multidimensional knowledge graph framework for e-learning,” *Knowledge-Based Systems*, vol. 195, pp. 105618, 2020.

[39] J. K. Tarus, Z. Niu and D. Kalui, “A hybrid recommender system for e-learning based on context awareness and sequential pattern mining,” *Soft Computing*, vol. 22, no. 8, pp. 2449–2461, 2018.

[40] S. Fatahi, F. Shabanali-Fami and H. Moradi, “An empirical study of using sequential behavior pattern mining approach to predict learning styles,” *Education and Information Technology*, vol. 23, no. 1, pp. 1427–1445, 2018.

[41] S. Bhaskaran and B. Santhi, “An efficient personalized trust-based hybrid recommendation (TBHR) strategy for e-learning system in cloud computing,” *Cluster Computing*, vol. 22, no. 1, pp. 1137–1149, 2019.

[42] C. Bhatt, M. Cooper and J. Zhao, “SeqSense: Video recommendation using topic sequence mining,” in *Proc. of Schoeffmann K.* (Ed) *MultiMedia Modeling, Lecture Notes in Computer Science*, Bangkok, Thailand, Cham, Springer, vol. 10705, pp. 252–263, 2018.

[43] Q. Chen, H. Zhao, W. Li, P. Huang and W. Ou. “Behavior sequence transformer for e-commerce recommendation in alibaba,” in *Proc. of the 1st Int. Workshop on Deep Learning Practice for High-Dimensional Sparse Data (DLP-KDD ‘19), Association for Computing Machinery*, Anchorage Alaska, United States, pp. 1–4, 2019.

[44] S. Park, N. C. Suresh and B. Jeong, “Sequence-based clustering for web usage mining: A new experimental framework and ANN-enhanced K-means algorithm,” *Data & Knowledge Engineering*, vol. 65, no. 3, pp. 512–543, 2008.