A Virtual Tutor to Enhance the Solving Skills of School Children Using Performance Evaluation and Navigation System

Shanta Sodur, Harmeet Singh, Rohan Pol, and Mohit Kale

Abstract The proposed *AI-based Performance Analysis and Navigation System* (AI-PANS) aims at the development of self-learning app for the students of Grade 1–10 students studying in the Maharashtra (India) State Board and also for the teachers. The proposed AI-PANS is designed for: – Database Generation, to ease the process of making new questions. – Recommend question paper based on the student's current and past performance in every iteration. Modify/redefine the *Degree Of Difficulty* (DOD) based on the time taken and correctness of the attempted question by a large set of students. – Navigate the student across topics based on his/her performance and in turn help him/her in clearing the concepts, improve the problem-solving skills, build confidence, retain interest, and prepare him/her for competitive exams. – Hand-holding in the form of curated solutions in the form of text, images, audio, etc., and summary of each topic. In the proposed system, we also proposed a novel mechanism to redefine the DOD assigned to every question in the beginning.

Keywords Clustering · Database generation · Education · Performance analysis · Question recommendation · Test papers

S. Sodur (✉) · H. Singh · R. Pol · M. Kale
Vivekanand Education Society's Institute of Technology, Mumbai, Maharashtra, India
e-mail: shanta.sondur@ves.ac.in

H. Singh
e-mail: 2018.harmeet.kathoda@ves.ac.in

R. Pol
e-mail: 2018.rohan.pol@ves.ac.in

M. Kale
e-mail: 2016.mohit.kale@ves.ac.in

1 Introduction

Mathematics have long held a high level of respect among their academic peers. Widespread fear of mathematics appears in many forms. If we are able to help a student in understanding the concept and develop interest in the subject by posing the problems which are simple and based on day to day activities along with the solution which is simple in words, correct and meaningful without any ambiguity, then the student gains interest and willingness in solving the next problem on his own. If the system is able to monitor his/her performance, then we can take the student either forward to the next level or else backward to the basic level (in a particular section).

2 Objectives

– To make mathematics easy to understand for students from basic to complex concepts.
– To create an application which will give *Multiple-Choice Questions* (MCQ) format tests to the students on various mathematical concepts, provide them with simple solution, and analyze their answers based on correctness and degree of difficulty.
– Free app for the school students of grade 5–10 with bilingual (English and Marathi) features for students of rural areas.
– To improve students' domain-general problem-solving skills, students' test-taking abilities, such as their time management.
– To create a recommendation system which will navigate the student to appropriate question tests according to his ability.
– To provide performance analysis at student level, class level, and school level.
– To help students and teachers to develop an interest and remove mathematics anxiety.

Machine learning and artificial intelligence (Supervised and unsupervised) are the primary domains for this project as they will be used for algorithm development, performance analysis, and students' navigation.

The scope of the project includes:

– Database Generation
– Question paper Recommendation
– Performance Evaluation
– Redefining the Degree Of Difficulty (DOD) and time of the Question
– Navigation

3 Literature Survey

3.1 Existing Applications

– **Khan Academy**: Khan Academy [1] It is an Interactive platform for students and teachers with over 4000 videos on educational topics like which includes maths.
– **Photomath**: PhotoMath [2] This is artificial intelligence-based application which uses 2 types of algorithms, namely *Optical character recognition* (OCR) and equation solving programs. User needs to click image of the written math problem, and application returns the solution.
– **Socrative Student**: Socrative Student [3] This is an application to conduct quizzes, surveys, etc.

3.2 Research Papers

Student Academic Performance Monitoring and Evaluation Using Data Mining Techniques suggests usage of data mining technique helps students expand their academic result [4].

Dynamic Question Paper Template Generation Using Bi-proportional Scaling Method elaborates on the need of constant evaluation to check progress of various cognitive skills under a subject at different stages of learning [5].

A Web-based testing system with dynamic question generation signifies the rising utilization of electronic media for education [6]

Although there has been extensive research in the field of Edtech, one significant limitation faced is the generation of questions. This often results in additional stress on the teachers. Instead of the manual method followed by a large number of applications, we decide to automate this process. Another limitation that stands out is the non-changing difficulty level of questions and the often overlooked performance of the student which results in a static system for the students to interact with.

Hence, we come up with the proposed application which aims to solve the shortcomings mentioned above.

4 Proposed Methodology

The proposed application is divided into five major parts. First part is database generation which generates the questions in an automated manner; second is question recommender for personalized recommendations; third is performance analysis responsible for analyzing the student's performance; then, fourth is the naviagtion module followed by DOD clustering module as the fifth part which dynamically changes the question's difficulty level based on relevant past data (Fig. 1).
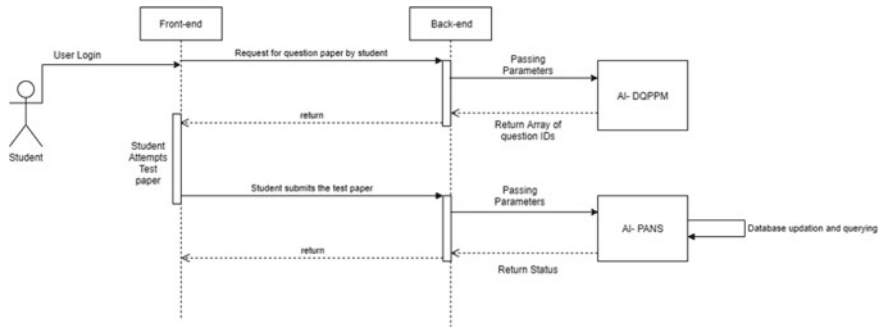
Fig. 1 Overview of the application

The above image shows the two major modules under the application. The *AI-based Dynamic Model for Question Paper Generation and Performance Monitoring* (AI-DQPPM) module includes the question recommender and navigation module. The second module is AI-PANS which consists of the performance analyzer and the DOD clustering sub-module.

4.1 Database Generation

This module consists of generating a database of varieties of problems under algebra vertical (Std V–X) using a generic problem statement. This database is generated through an algorithm designed for a generic problem with all sorts of possible variations (along with solutions), and also with different degrees of difficulties which will help the student of any category (Excellent,above average,average, below average) to clear his/her concepts from basic to advanced level. These variations include the questions of MCQ type (single correct, multiple correct, true/false, match the pair, and fill in the blanks). The questions and solutions are in the form of text, images, videos, and audios.

4.1.1   List of Sections for Algebra Vertical

– 0301 Equation
– 0302 - /, Using letter in place of a number—variable/unknown, solution to equation in variables
– 0303 Mathematical Expressions
– 0304 Algebraic Expressions

4.2   Question Recommender

Recommends personalized questions to the student based on his performance on individual questions, easiest question from the topic, and questions from students having similar performance.

This component contains three standalone modules that each recommend questions based on a different criteria.

– **Popularity-Based Recommendation**: This module queries the database and picks out the easiest, i.e., questions that have a high chance of being ticked correct by the student from that particular module. This module is useful when the student has just registered and hasn't yet attempted a lot of questions.
– **User based recommendation**: This module uses the student's performance to pick out questions according to it, with the goal of increasing his/her performance after each iteration of a question paper. This module helps the student by recommending them the questions they might not be doing well in.
– **Collaborative Filtering**: The main task of this module is to recommend newer questions to the student which he hasn't attempted yet. It achieves this by finding out other similar student to him/her in performance and then recommending the questions they have performed well in.

4.3   Performance Analyzer

Measures the performance of the student based on correctness, time taken, and DOD of the questions considering past history. For every correct attempt, the performance of the student is incremented proportionally to the difficulty level the question had, and the time he/she took to answer the question. The same results in a decrement in performance level if the attempt goes wrong. Also, the decrement in performance is directly proportional to the time taken to solve the question.

This parameter will be used to calculate the student's performance for each variation the student attempts and use it as a marker to judge his current proficiency level in the corresponding topic.

The above image also shows the DOD clusterer module which will be discussed further in this section.

4.4   Navigation

The main function of this module is to determine whether the student should be promoted or demoted to the next or previous module based on his performance on the current topic. If the student's performance surpasses the median value of 4 in the

current topic, then this module will move him to the next respective topic. Similarly, when his/her median performance in the current topic goes below 1, the module will demote him to the previous respective module as the student might need some more practice to attempt the questions of the current topic.

4.5 DOD Clustering Module

This module clusters all the variations in 5°C of difficulty (DOD) ranging from 1 to 5. The clustering algorithm is implemented on 2 levels. In the first level, all the variations are are clustered using the *Corrections Ratio or Success Ratio*, defined as

$$\text{Correction Ratio} = \frac{\text{Correctly Attempted}}{\text{Attempted}} \qquad (1)$$

where CorrectlyAttempted is the number of times the question was attempted correctly and *Attempted* is the number of time the question is attempted.

In the second level, the variations are clustered according to the *time taken* by the student; hence, the clusters created are called as *time based clusters or T-clusters*. The *T-clusters* range from 1 to 5. If a variation has *T-cluster* of 1, then the time taken by students to solve that variation is very less than expected. Similarly, if *T-cluster* of a variation is 5, then time taken by students to solve that variation is very high. So, in conclusion, the variations with *T-cluster* of 1 are considered easier, and the variations with *T-cluster* of 5 are considered difficult. The proportional difference between expected time and actual time taken is used for *Time based clustering*, and it is calculated using the formula given below

$$\text{Expected Time} = \text{Correctly Attempted} * \text{Time Assigned} \qquad (2)$$

where Time Assigned, the time expected by the professionals that the student should take to successfully solve the question.

$$\text{Proportional Difference} = \frac{\text{Time Taken} - \text{Exptected Time}}{\text{Expected Time}} \qquad (3)$$

where Time Taken is the cumulative time taken by all the students to attempt the question correctly.

After dividing variations in 5 clusters using both the methods, the final difficulty level is assigned using swapping method. For example, If a variation with id 101 has correction ratio-based cluster as 3 and it has T-cluster as 5, then that variation will be assigned DOD of level 4. Whereas if the same variation had the T-cluster of 1, then the DOD 2 should be assigned to it.

Fig. 2 Snapshot of Excel sheet generated in module 030101

5 Implementation

This section discusses the implementation part of our proposed system. First, we show the outcome of coep-package developed to standardize the question generation process. Next, we show the integration of the proposed system with a Web site through APIs along with the performance analysis and navigation part.

5.1 CSV Generation

See Fig. 2.

5.2 Navigation

This section shows how the student will be navigated across the sections/subsections in the current topic (Fig. 3).
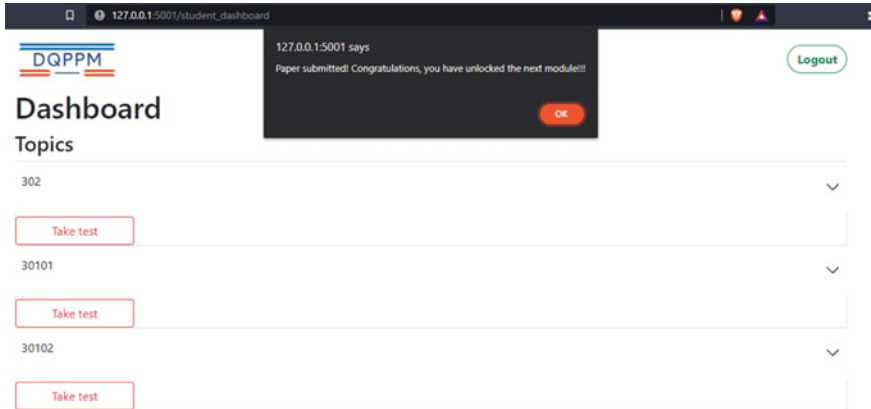
5.3 DOD Clustering

See Fig. 4.

Fig. 3 Students being promoted after she/he reaches the performance of 4.5

CorrectlyAttempted	Attempted	Total_Time	clusters	time_clusters	suggested_time_in_seconds	suggested_dods
993	1520	179371	4	1	54	3
1264	1798	222855	2	2	73	2
1142	1703	204393	1	3	78	1
1082	1609	192064	3	3	80	3
991	1527	180685	4	3	78	4
1008	1532	180406	4	4	86	4
1095	1641	188560	3	3	79	3

Fig. 4 Snapshot after executing DOD clustering algorithm

6 Results

We present our results simulated through an algorithm which attempts the question paper randomly for a set of 30 iterations See Fig. 5.

Finally, we infer that we have simulated the solving process of the student anticipating the student may follow this pattern.

However, the exact results would be available only when the question paper is rendered to the students of respective grade.

7 Conclusion

We conclude that, we have successfully implemented the proposed algorithm and tested the performance through simulated results against the expected ones.

From the results, we can see that we have automated the CSV generation with multiple questions populated based on the requirement, through our package *coep-*

Fig. 5 Graph of
Performance against No. of
iterations for 30 question
papers
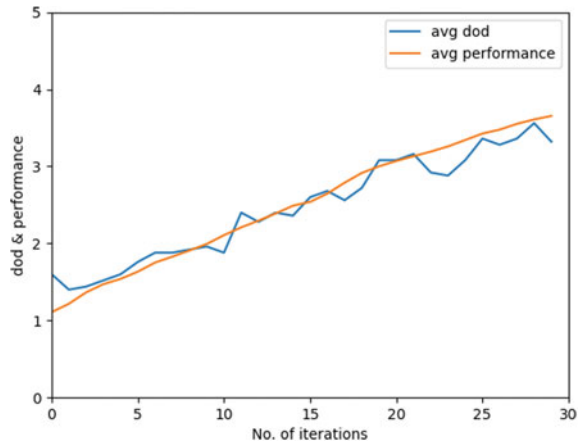
package. We also devised the technique for DOD clustering, and finally, we have
demonstrated the navigation of student based on his/her performance.

References

1. Khan Academy. Accessed 29 Mar 2022. https://www.khanacademy.org
2. PhotoMath. Accessed 29 Mar 2022. https://photomath.com/en/
3. Socrative Student. Accessed 29 Mar 2022. https://www.socrative.com/
4. Ogor EN (2007) Student academic performance monitoring and evaluation using data mining
   techniques. In: Electronics, Robotics and Automotive Mechanics Conference (CERMA 2007),
   pp 354–359. https://doi.org/10.1109/CERMA.2007.4367712
5. Paul DV, Pawar JD (2013) Dynamic question paper template generation using Bi-proportional
   scaling method. In: 2013 IEEE fifth international conference on technology for education (t4e
   2013), pp 80–83. https://doi.org/10.1109/T4E.2013.28
6. McGough J, Mortensen J, Johnson J, Fadali S (2001) A web-based testing system with dynamic
   question generation. In: 31st annual frontiers in education conference. Impact on Engineering
   and Science Education. Conference Proceedings (Cat. No.01CH37193), pp S3C-23. https://doi.
   org/10.1109/FIE.2001.964056