I.J. Modern Education and Computer Science, **2023, 2, 54-61**
Published Online on April 8, 2023 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijmecs.2023.02.05

A Performance Analysis of the Impact of Prior-Knowledge on Computational Thinking

Swanand K. Navandar*
Dr. Babasaheb Ambedkar Technological University, Lonere, Raigad, 402103, India
E-mail: swanandnavandar@gmail.com
ORCID iD: http://orcid.org/0000-0003-2986-3427
*Corresponding Author

Arvind W. Kiwelekar
Dr. Babasaheb Ambedkar Technological University, Lonere, Raigad, 402103, India
E-mail: awk@dbatu.ac.in
ORCID iD: http://orcid.org/0000-0002-3407-0221

Manjushree D. Laddha
Dr. Babasaheb Ambedkar Technological University, Lonere, Raigad, 402103, India
E-mail: mdladdha@dbatu.ac.in
ORCID iD: http://orcid.org/0000-0002-6265-1728

Received: 20 June, 2022; Revised: 25 July, 2022; Accepted: 21 August, 2022; Published: 08 April, 2023

Abstract: Previously acquired knowledge plays a significant role to learn new knowledge and skills. Previously acquired knowledge consists of Short-term memory and Long-term memory. Though it is a well-accepted learning phenomenon, it is challenging to empirically analyse the impact of prior knowledge on learning. In this paper, we use two systems models for human thinking proposed by Nobel Laureate Prof. Daniel Kahneman. This is a model for human cognition which uses two systems of thinking—the first being quick and intuitively known as fast thinking and the second being slow and tedious known as slow thinking. While slow thinking uses long-term memory, fast thinking uses short-term memory. The impact of prior knowledge of programming language is analyzed to learn a new programming language. We assigned a learning task to two different groups with one having learnt a programming language i.e. senior students and the second group without any prior knowledge of programming language i.e. freshers. The impact of prior knowledge is measured and compared against the time taken to answer quizzes.

Index Terms: Previously acquired knowledge, performance, Kahneman two system model.

1. Introduction

The COVID-19 pandemic, and the worldwide movement toward online classrooms, have highlighted the importance and potential of effective remote learning environments. All examinations are conducted online, due to COVID-19. For assessments, instructors create a multiple-choice question(MCQ) and a Short answer question(SAQ). Courses are delivered and exams are administered via a variety of online platforms. There is a lot of data gathered from student responses. Different problems linked to students can be solved by in-sighting the data using statistics and Machine Learning approaches, such as student performance, course understanding, sequence of content studied by students, and even placement[1] of students depending on technical talent.

Prior knowledge is a key aspect of learning programming, mathematics, and other ideas. Because of socioeconomic variations, knowledge influences toddlers' word learning and comprehension[2]. Prerequisite skills are fundamental concepts or information required to solve or implement a challenge. Sometimes the abilities are required for the same course, or skills from an introductory course are required for an advanced course.

Students with no prior programming experience are reported to suffer more than students with prior programming expertise[3]. When it comes to MCQ and SAQ-type questions, students tend to utilize a fast-thinking technique rather than a slow-thinking strategy[4], which results in incorrect answers.

Decision-making processes confront the brain daily, according to neuropsychology. Many factors influence decision-making, ranging from biological cues to reward assessments. In these decision-making situations, many priming factors may be present. There is a need to understand the role of previously acquired knowledge and student performance.

In the quiz, 40 students voluntarily participated from both years, first and third. For the correct collecting of data, willingness is more crucial. The test, which is conducted for performance analysis, consists of 20 questions. The questions contain both multiple-choice questions and short answers. For Fast-thinking and Slow-thinking separately, the same quiz with the questions shuffled is conducted with various timestamps. Fast thinking has a lower timestamp value than slow thinking. Students' responses to both strategies were recorded. Each correct answer receives one mark, while incorrect answer receives zero marks.

Section 2 of the article defines Kahneman's two-system model of thinking. Section 3 on the literature survey Section 4 examines the research problem. Section 5 is related to data collection, and Section 6 validates dependencies. The paper concludes with a path for future work.

2. Kahneman's two-systems model of thinking

Kahneman's book[5] aims to provide a fresh viewpoint on how the mind works, based on current advances in cognitive and social psychology. To describe mental functioning, he employs the concept of two systems, System 1 and System 2, which produce fast (intuitive) and slow (deliberate) thinking, respectively. System 1 is mostly unconscious, and it makes snap decisions based on our emotions and previous experiences. System 2 is rational and conscious. They work together to give us a perspective on the world[6,7,8,9,10].

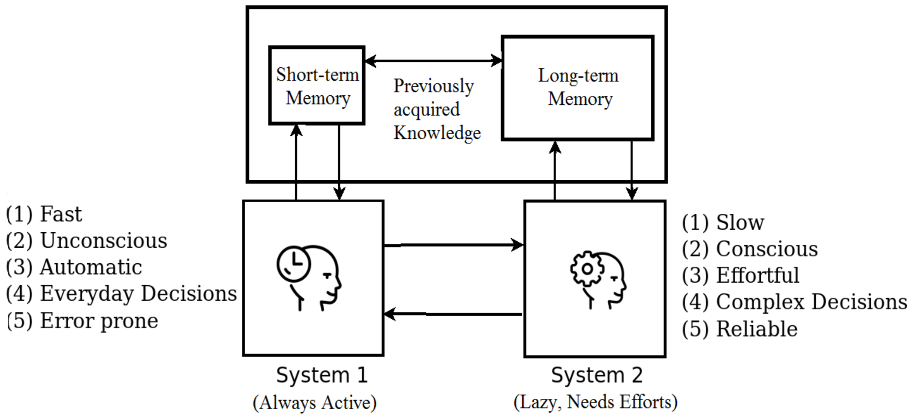

Fig. 1. Kahneman's two-systems model of thinking

As shown in Fig.1, Short-term memory(STM) stores frequently performed operations from long-term memory. This STM can be increased by continuous and dedicated practice of programming. System 1 always uses readily accessible short-term memory. System 1 is Fast-thinking and is always active. System 2 is Slow-thinking and is inherently sluggish, therefore it takes time to activate it. System 2 always makes use of existing knowledge in long-term memory. Long-term memory and short-term memory both store previously acquired information.

Fast thinking is both natural and unconscious, yet it can lead to a wide range of cognitive biases. While this cognitive bias can help us make speedy decisions, it can also hinder our ability to reach rational conclusions.

To appropriately answer questions, the following examples demonstrate the presence and application of two different thinking styles.

Example 1: We rapidly and accurately answer the question 1 + 3 =? without exerting any effort. However, to answer the question 18 X 24 =? we must exert effort and perform deliberate calculations, which necessitates slow thinking.

Example 2: What is the return type of the given function? as stated in Fig,2,

```
int fun(float aa)
{
        return((int)aa);
}
```

Fig. 2. definition of fun function

Anyone familiar with the C programming language's concepts of function can respond quickly and correctly.

Example 3: What does the function shown in Fig. 3 return? when the following input data array is called: [31, 71, 21, 91, 14]

```
int getMin(int data[])
{
        int min=0;
        for(i=0; i<5; i++)
        {
            if (data[i]<min)
            {
                    min=data[i];
            }
        }
        return min;
}
```

Fig. 3. Definition of a getMin function

Students answer the question with 14, which is erroneous since they fail to notice the program and see that the name of the program is getMin, as shown in Fig 3. The right response for a given program is 0. We can see how thinking fast in this situation results in choosing the incorrect course of action because the min variable is set to zero. Poor judgment might result from cognitive bias brought on by fast thinking. In the case of Slow thinking, students see every step deliberately so can easily find the mistake that the program will return zero instead of 14. As zero is initialized to min variable so while comparing with an array element, zero is always less than that of array elements. The return value for the getMin function is zero.

In Example 1, a simple addition operation is asked, which we frequently use in day-to-day life but multiplication of two digit number required slow, dedicated thinking. Similarly, Example 2 shows that a simple return type of function is asked, which concepts we frequently use in the case of C-programming, so short-term memory stores this information and is readily available when required. However, in the case of Example 3, a simple comparison operation is done, but here the min variable is set to zero, which is the main trick used so that swapping is not done as the minimum value is zero.

As a result, both Examples 1 addition part and Example 2 can be correctly solved using System 1. However, in Example 1, to solve the multiplication part and to recognise the trick used in Example 3, slow thinking is required, i.e., System 2.

3. Literature Survey

One of the most important factors in learning programming, mathematics, and other concepts is prior knowledge. The capacity of youngsters to learn vocabulary from storybooks may be influenced by their prior knowledge. Children with greater vocabularies are more likely to learn vocabulary implicitly through storybooks than children with lesser vocabularies, according to a previous study[11,12]. Knowledge has an impact on preschoolers' word learning and comprehension due to socioeconomic differences[2]. In the medical area also, pre-existing knowledge levels could be a good predictor of patient success and adherence to internet-based cognitive behavior therapy[13]. When it comes to developing higher-order mathematical thinking, students' past mathematical knowledge is crucial[14].

Table 1 shows the statistical and machine learning methods that were applied. The aims of various statistical tools differ, yet they are all related to predicting students' performance.

Classification and regression algorithms are used to identify trends in study-related data as well as data on students' social behavior to predict how well students will perform on final exams[15]. Based on the prior achievements of similar pupils, collaborative filtering techniques are used on this data to forecast the final grades.

The association between prerequisite competency and student success in an upper-division computing course is determined using correlation analysis. Students' performance is based on their mastery of the prerequisite skills[16]. Additionally, it was discovered that, with the aid of correlation analysis, retention performance may be predicted and is corelated to prerequisite skills[17].

Data from prerequisite skills is used to determine how well students do in post-requisite skills. This aids in honing the pre-requisite abilities needed to learn the post-requisite skills[18].

The mastery of pupils in specific knowledge is tracked with the help of the Q-matrix of educational prior[19], which aids in many educational applications like training particular knowledge or exercise recommendations. Exercise aids in monitoring students' post-requisite skills as well. Prerequisite structure of knowledge components aids in lesson planning and mastery assessment[20].

In this literature survey, the majority of research is conducted on the required and post-requisite skills to predict student performance. Depending on the nature of the data set and the nature of the problem, several statistical methods are applied. Our goal is distinct from that of this research. Our research goal is to correlate the year of students and their performances with the thinking process used, i.e. fast or slow thinking.

The student's year and the performance of students are two categorical variables we used for experiments, so a Chi-square test is used to find the correlation between two categorical variables.

Table 1. Literature Survey

Sr. No.	Statistical Method	Purpose	Source
1	Classification, regression	To forecast a student's exam success or failure, as well as their final grade	[15]
2	Correlation	The correlation between final exam performance and prerequisite skills was substantial.	[16]
		Using prerequisite skill traits to predict retention performance	[17]
3	Linear Regression	Using prerequisite skills data to predict student success on post-requisite skills	[18]
4	Reference distance (RefD)	To determine the relationship between Computer Science and Math courses as prerequisites.	[21]
5	Regression and correlation	Predicting prerequisite links between corresponding course units using heterogeneous content sources such as quizzes and textual course material	[22]
6	Q-matrix	To track students' post-requisite skills from a specific knowledge point of view through exercises.	[19]
		Identifying necessary structure based on diversity in student performance on an assessment.	[20]

4. Research Problem

There is a large difference in previously acquired knowledge between first-year and third-year pupils. As a result, we're curious to discover how prior knowledge and the sort of thinking process (Fast or Slow) influence students' programming performance.

To validate the year and performance dependencies concerning fast and slow thinking, we separately validate the same hypothesis concerning students' problem-solving strategy, i.e. fast or slow thinking as

H0: *There is no association between year and performance of student*

H1: *There is an association between year and performance of student*

5. Dataset

Students' quiz responses are collected using Google forms, which creates a dataset for experimentation based on the quiz approach, i.e., fast or slow thinking. For the quiz, fast-thinking is given less time than slow thinking. The quiz consists of four short answer questions and sixteen multiple-choice questions. Each question is worth one point. The student who provides the correct answer receives one point, while the student who provides the incorrect response receives zero points. As a result, the dataset is made up of the sum of the scores of all students.

6. Research Methodology

Students solve the quiz using a fast-thinking approach [4] by default so that their performance is reduced. So we are interested in finding the impact of prior knowledge on students' performance for both thinking approaches.

The first step is to collect the responses of students with the help of a Google form based on thinking approach use. For fast thinking, the quiz timer is less than that of the slow thinking approach.

Students' performance is calculated in the second step for both thinking approaches. For every correct answer, one mark is added to the students' score. Based on these scores, the performance of students is categorized into four categories: poor, average, good, and excellent.

In step three, with the help of the chi-square test, is used to analyse the correlation between the year of students and the performance of students for both types of thinking approach.

7. Validating Dependencies

Responses are examined to validate the year and performance dependencies of pupils. Because the dataset contains categorical data, the Chi-Square Test is used to find a correlation. The score is derived from the data, which consists of student replies. As data consists of students' responses that score.

Categories the students score into poor, average, good, and excellent. A score ranging from 0 to 5 is considered a poor score. A score ranging from 6 to 10 is considered an average score. A score ranging from 11 to 15 is considered a good score. A score ranging from 16 to 20 is considered an excellent score.

Table 2 and Table 3 show that the Student Response Categories (X) and Year of Student (Y) are both categorical variables for Fast thinking and Slow thinking approaches respectively. A Chi-square test is used to determine the association between them.

A Chi-square test is used to validate the hypothesis given above. Table 4 and Table 5 show the results of the Chi-square test for the Fast thinking and Slow thinking approach respectively.

A contingency table, as shown in Tables 2 and 3, is produced to examine the relationships between the performance of first and third-year students using the thinking fast and slow approach. The graphs depict the observed frequencies in the dataset as well as the expected frequencies obtained using the Chi-square test. All questions are multiple-choice or short answer questions, and wrong answers degrade student performance while correct answers improve the score by one mark. A quiz has a total of 20 questions.

The Python program was created for this purpose by calculating Chi-square with the chi2 module from the Scipy.stat library.

Table 2. Contingency table for Fast Thinking approach

Student Response	Poor	Average	Good	Excellent
Observed Frequencies				
First Year	15	22	3	0
Third Year	2	26	11	1
Expected Frequencies				
First Year	8.5	24	7	0.5
Third Year	8.5	24	7	0.5

Table 3. Contingency table for Slow Thinking approach

Student Response	Poor	Average	Good	Excellent
Observed Frequencies				
First Year	1	22	11	6
Third Year	0	5	21	14
Expected Frequencies				
First Year	0.5	13.5	16	10
Third Year	0.5	13.5	16	10

Table 4. Results of Chi-square test for Fast Thinking approach

Probability	0.95	Critical Value	7.8147278
Statistic	15.845938	p-value	$0.00122 < 0.05$

Table 5. Results of Chi-square test for Slow Thinking approach

Probability	0.95	Critical Value	7.8147278
Statistic	18.028704	p-value	$0.000434 < 0.05$

The calculated Chi-square statistic for the Fast thinking approach ($x^2 = 15.8459$) is substantially more than the preset critical value of 7.8147, hence the null hypothesis, i.e. (H0) There is no association between year and performance of students is rejected. Also, the p-value is less than the significance level of 0.05, indicating the results are statistically significant as shown in Table 4. In other words, there is a statistically significant dependency between year and performance of students is observed.

The calculated Chi-square statistic for the Slow thinking approach ($x^2 = 18.0287$) is substantially more than the preset critical value of 7.8147, hence the null hypothesis, i.e. (H0) There is no association between year and performance of students is rejected. Also, the p-value is less than the significance level of 0.05, indicating the results are statistically significant as shown in Table 5. In other words, there is a statistically significant relationship between the academic year and student achievement.

We may conclude from Tables 4 and 5 that student performance is statistically significant concerning the year of students in both fast thinking and slow thinking approaches. As a result, we can say that Students' thinking processes are influenced by previously acquired knowledge.

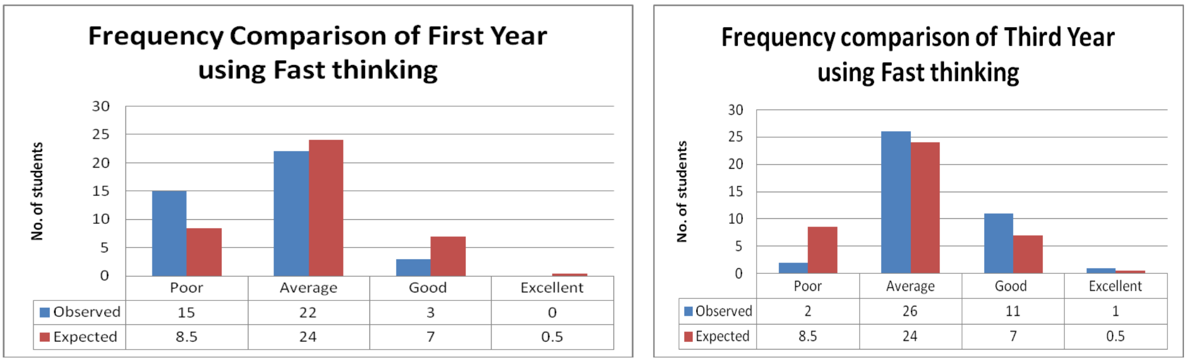

Fig. 2. Observed and Expected Frequencies using Fast thinking approach

Fig.2. compares the expected and observed frequencies using the Fast thinking approach. In the case of first-year students, the observed frequency for poor scores is higher, and the observed frequencies for average, good, and excellent scores are lower than expected. In the case of third-year students, the observed frequency for poor scores is lower than expected, while the observed frequency for average, good, and excellent scores is higher than expected. As a result, we may conclude that in the case of fast thinking, the number of poor performance scores is higher in the first year than in the third year. Third-year students outperform first-year students in terms of average, good, and excellent achievement. Although no student received excellent grades in the first year, at least one did so in the third year while using the Fast thinking approach.

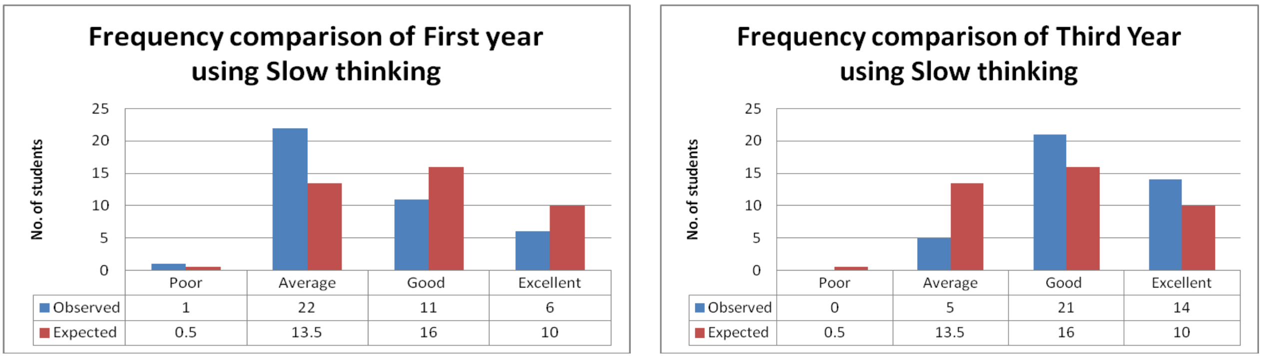

Fig. 3. Observed and Expected Frequencies using Slow thinking approach

Fig.3. compares the expected and observed frequencies using the Slow thinking approach. In the case of first-year students, the observed frequency for poor and average scores is higher. The observed frequencies for good and excellent scores are lower than expected. In the case of third-year students, the observed frequency for poor scores is null. The observed frequency for average score is lower than expected. The observed frequency for good and excellent scores is much higher than expected. As a result, we can infer that in the case of Slow thinking, there was just one first-year poor performance score and none in the third. In terms of good and excellent achievement, third-year students exceed first-year pupils. One first-year student who used the slow thinking strategy scored poorly, while third-year students outperformed first-year students.

8. Limitation and Threats to Validity

This paper is limited as we only studied the performance in C-Programming concepts of first and third-year students. Without a proctor, the Assessments are taken online. The risk is that students will open a different tab and execute the program using an online compiler to provide their answer, where manual solving of program output is expected. MCQs and SAQs are present in the quiz. There may be an opportunity to ask friends for assistance. Since there is no deduction for incorrect answers on MCQs and SAQs, there is a risk that the answers can be guessed.

To overcome this limitation, the quiz should be run for different programming languages as well as in offline mode with a proctor.

9. Conclusion and Future Work

This paper presents the year and performance dependencies of students concerning fast and slow thinking through C-programming concepts.

The null hypothesis (H0), which states that there is no correlation between a student's year and performance, is rejected in the instance of the fast thinking technique since (x2 = 15.8459) is significantly higher than the predetermined critical value of 7.8147. Additionally, the results are statistically significant because the p-value is under the threshold of 0.05.

When applying the slow thinking strategy, the null hypothesis (H0), which claims that there is no association between a student's grade and performance, is disproved because (x2 = 18.0287) is significantly higher than the set critical threshold of 7.8147. The results are statistically significant because the p-value is less than the threshold of 0.05.

There is significant variation between previously acquired knowledge of first-year and third-year students. From these findings, we can claim that the performance of students is reliant on previously acquired knowledge. In other words, for both types of thinking processes, there is a statistically significant relationship between year and performance of students.

In the current study, we only take into account MCQs and SAQs for quizzes on C-programming topics; however, in the future, we may also evaluate students' performance with other programming languages. we're also interested in finding quizzes on topics like writing a piece of code, fixing code for intended results, and rearranging programs from supplied sets of code. Additionally, this will test their programming abilities.

References

[1] M.D. Laddha, V.T. Lokare, A.W. Kiwelekar, L.D. Netak "Performance analysis of the impact of technical skills on employability", International Journal of Performability Engineering vol. 17, 2021

[2] Kaefer, Tanya, Susan B. Neuman, and Ashley M. Pinkham. "Pre-existing background knowledge influences socioeconomic differences in preschoolers' word learning and comprehension.", Reading Psychology, vol. 36, 2019, pp 203-231.

[3] H. Burd, A. Bell, E. Hemberg and U.M. O'Reilly, "Analyzing Pre-Existing Knowledge and Performance in a Programming MOOC". In Proceedings of the Seventh ACM Conference on Learning@ Scale Aug 2020, pp. 281-284.

[4] S. K. Navandar, M. D. Laddha, A. W. Kiwelekar, " Analyzing Performance in a Computer Programming Course Through a Two-System Model", International Journal of Performability Engineering, vol.18, 2022.

[5] D. Kahneman, "Thinking, fast and slow", 2017.

[6] A.W. Kiwelekar, S. Navandar, D.K. Yadav, "A two-systems perspective for computational thinking", 12th International Conference on Intelligent Human Interaction (IHCI 2020), 2020.

[7] D. Chen, Y. Bai, W. Zhao, S. Ament, J. M. Gregoire, C. P. Gomes, "Deep reasoning networks: Thinking fast and slow", arXiv preprint arXiv:1906.00855, 2019.

[8] J.S.B. Evans, K.E. Stanovich, "Dual-process theories of higher cognition: Advancing the debate", Perspectives on Psychological Science, Vol.8, 2013, pp.223-241.

[9] G. Goel, N. Chen, & A. Wierman, "Thinking fast and slow: Optimization decomposition across timescales", In 2017 ieee 56th annual conference on decision and control (cdc), 2017, pp. 1291–1298.

[10] U. Kannengiesser, J.S. Gero, "Design thinking, fast and slow: a framework for kahneman's dual-system theory in design", Design Science, Vol.5, 2019, pp.1-21.

[11] C. B. Leung, "Preschoolers' acquisition of scientific vocabulary through repeated readaloud events, retellings, and hands-on science activities", Reading Psychology, vol. 29, 2008, pp 165–193. doi:10.1080/02702710801964090

[12] S. Stahl, & W. Nagy, "Teaching word meanings", Mahwah, NJ: Erlbaum, 2006.

[13] J. Linardon, J. Broadbent, A. Shatte, and M. Fuller-Tyszkiewicz, "The role of pre-existing knowledge and knowledge acquisition in internet-based cognitive-behavioural therapy for eating disorders", Computers in Human Behavior, 2022., p.107332.

[14] S. Encheva, January, "Dependencies between Pre-existing and Newly Obtained Knowledge", In 2016 International Conference on Artificial Intelligence: Technologies and Applications, Atlantis Press, 2016, pp. 121-123.

[15] H. Bydžovská, "A comparative analysis of techniques for predicting student performance", International Educational Data Mining Society, 2016.

[16] S. Valstar, W.G. Griswold, L. Porter, "The relationship between prerequisite proficiency and student performance in an upper-division computing course", In: Proceedings of the 50th ACM Technical Symposium on Computer Science Education. 2019, pp. 794–800.

[17] X. Xiong, S. Adjei, N. Heffernan, "Improving retention performance prediction with prerequisite skill features", In: Educational Data Mining, 2014, Citeseer 2014.

[18] S.A. Adjei, A.F. Botelho, N.T. Heffernan, "Predicting student performance on post-requisite skills using prerequisite skill data: an alternative method for refining prerequisite skill structures", In: Proceedings of the sixth international conference on learning analytics & knowledge 2016, pp. 469–473.

[19] Y. Chen, Q. Liu , Z.,Huang, L.,Wu, E., Chen, R., Wu, Y.,Su, G, Hu, : "Tracking knowledge proficiency of students with educational priors", In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. pp. 989– 998, 2017.

[20] R. Scheines , E. Silver, I.M.Goldin, "Discovering prerequisite relationships among knowledge components", In: EDM, pp. 355–356, 2014.

[21] C. Liang, Z. Wu, W. Huang, C.L.Giles, “Measuring prerequisite relations among concepts”, In: Proceedings of the 2015 conference on empirical methods in naturallanguage processing, 2015, pp. 1668–1674.

[22] D.S. Chaplot, Y. Yang, J. Carbonell, K.R. Koedinger, “Data-driven automated induction of prerequisite structure graphs”, International Educational Data Mining Society, 2016.

Authors’ Profiles

Swanand K. Navandar received his Bachelor of Engineering degree in Information Technology from Savitribai Phule Pune University and Master of Technology degree in Computer Engineering from Dr. Babasaheb Ambedkar Technological University. He is pursuing his Ph.D. degree from Dr. Babasaheb Ambedkar Technological University, lonere, India.

His area of interest includes Cognitive Science, Artificial Intelligence, Data mining, Machine learning, Algorithms, Internet of Things and Human-computer Interaction, etc. He can be contacted at swanandnavandar@gmail.com.

Arvind W Kiwelekar is a Professor in the Department of Computer Engineering at Dr Babasaheb Ambedkar Technological University-Lonere, India. He is the founding head of the department. He has thirty years of teaching and research experience. His areas of interest include diverse topics, including Artificial Intelligence, Blockchain Technology, Information and Communication Technologies for Sustainable Development (ICT4SD), Learning Analytics, Ontology, and Software Architecture. He published several papers at leading venues such as ECSA, IJCAI, ICSE, CompSAC and SAC. He graduated with a PhD from Indian Institute of Technology Bombay (2012), M E from Veermata Jijabai Technical Institute (VJTI) Mumbai(1997) and B E from Marathwada University (1991). He received a research fellowship from the Indian Academy of Sciences Bangalore, Ministry of Education Govt. of India and IBM. Alumni of the Department of Computer Engineering. at DBATU has constituted an award to honour his dedication towards teaching

Manjushree D. Laddha received the Ph.D degree (Doctor of Philosophy) in Computer Engineering from the Dr. Babasaheb Ambedkar Technological University, Lonere, India, with the Dissertation “A Software Architecture Perspective of Learning Analytics”. She is a Assistant Professor of Computer Engineering at Dr. Babasaheb Ambedkar Technological University, Lonere, India. In addition, she is serving as researcher in the research group on Cognitive Science and Learning Analytics. She has published journals papers, books chapters, and papers in the International Conferences. Her research interests are in Machine Learning, Cognitive Science, Learning Analytics, and Software Engineering

How to cite this paper: Swanand K. Navandar, Arvind W. Kiwelekar, Manjushree D. Laddha, "A Performance Analysis of the Impact of Prior-Knowledge on Computational Thinking", International Journal of Modern Education and Computer Science(IJMECS), Vol.15, No.2, pp. 54-61, 2023. DOI:10.5815/ijmecs.2023.02.05