How Well Students Perceive Their Understanding of Logic Programming Course Content?

S. Lovrenčić, V.Sekovanić
University of Zagreb Faculty of Organization and Informatics, Varaždin, Croatia
sandra.lovrencic@foi.unizg.hr
vlatka.sekovanic@foi.unizg.hr

***Abstract* - Teaching and learning programming languages is challenging in many ways. Students are often more familiar with imperative paradigm, and therefore programming languages that belong to declarative paradigm are even more demanding. Logic programming languages are usually taught when students already have prior knowledge of programming concepts and logic. This should make their learning easier, and also their perception to which level they have mastered course content prior to exam should be quite good. The paper describes analysis and results from five-year research on a course where logic programming is taught in the Prolog programming language, and includes one year of online teaching. Students' feedback on their perception of understanding the course content, especially practical part is analysed and compared to their success in passing the course at the end of the semester. Differences between perception and success are discussed together with results of previous research of learning problems perceived by students, with the aim of investigating the role of opinions and perception in learning analytics and improving the learning design of this type of course.**

Keywords - logic programming course; learning analytics; perceived understanding

I. INTRODUCTION

Logic programming, as part of declarative programming paradigm, is often taught at computer science study programmes as compulsory or elective course. Declarative programming paradigm is considered as contrast to imperative programming paradigm, which has procedural and object-oriented programming as most known representatives. Contrast relates to the different way of encoding information and procedures. For example, logic programming is based on logic and is mostly used for representing declarative knowledge (facts and rules) with reasoning (search) procedures already defined in the programming language [1]. It is descriptive by nature and programmers' task is to describe chosen domain in a knowledge base by entering what is true [2].

Programming languages belonging to imperative programming paradigm, such as C or C++, or multiparadigm programming languages, for example Python, are often taught at introductory programming courses in the first year of undergraduate study. Logic programming languages, whose most prominent representative is Prolog [3,4], language that is based on first-order predicate logic [5], are taught at higher years. Knowledge of those languages continues to be very important - Prolog is today widely used in artificial intelligence field [6,7,8,9,10] and it is well-known that it was used for programming the parts of the the IBM's Watson supercomputer [11]. Therefore it is important to ensure that students understand the theoretical concepts important for understanding and learning logic programming.

Since logic programming is taught when students already have prior knowledge of programming concepts from previous programming courses and basic knowledge of logic from mathematics course, this would imply that they can learn new programming languages easier, but it is also understandable that this may not be the case if they belong to different programming paradigm. However, it would be expected that they can relatively correctly perceive whether they understand the course content well. The purpose of the research presented in this paper is to investigate whether this supposition is true, which can be done by comparing students' perception of understanding the course content with their success in passing the course. Results can also help in better understanding how student perception can be used in combination with learning analytics and help in learning design.

The paper is organized as follows. In Section II is reviewed previous research on the problems in teaching and learning programming in general as well as logic programming and the use of learning analytics in programming education. The research is described in Section III and results of the student survey during five year period are presented in Section IV, focusing on how students perceive their understanding of the course content in general and aditionally practical part, compared to their grades. In Section V are presented concluding remarks.

II. BACKGROUND

Programming knowledge is today one of the most important skills for prospective career. It is also one of the skills that is not easy to teach and learn, because students often have problems with understanding programming concepts, feel fear or anxiety and have lack of motivation [12,13,14]. They have different learning styles (for example, alone or in groups), which combined with the choice of programming language and the fact that programming is actually a set of skills and involves more processes, can make teaching and learning very difficult [15]. Important elements in teaching programming to beginners are curricula (what to teach), pedagogy (managing teaching and learnign process), choice of programming language and type of tool for chosen language (for example, visualisation tool or programming support tool) [16].

Since logic programming languages demand different approach to programming process than programming languages that students are already familiar with, above mentioned problems in teaching and learning do not become easier with students' prior experience. Most of the research about problems of teaching logic programming is based on experience with Prolog, since it is the most known and used logic programming language, with common conclusion that the different programming paradigm and its concepts are difficult to learn [17, 18, 19, 20], and therefore the same elements in teaching are important as in beginner courses [16], with balancing theoretical abstract and concrete practical approach, taking into consideration different learning styles [19] and structuring the teaching material properly [17].

Learning analytics is a discipline that is gaining moment and growing in potential for use in education for more than ten years, enabling collection, analysis and understanding the data about learners (students) with the goal to improve (optimize) the learning process [21]. Various methods of data collection and analysis as well as tools can be used in different areas of application, such as modelling user knowledge or domain, personalization or trend analysis [21, 22]. The use of learning analytics enables better understanding of students' learning process and can improve learning support and teaching [23], as well as retention of students [24]. Since it emerged, the discipline was also used to understand the learning of programming [25] with research oriented to tracking students' activity and comparing it to tinkering and changes in quality of programs [26] or tracking students' progress in code development and exploring the correlation with course performance and grades [27]. It can also be valuable tool in detecting students that have risk of failing some assignments or the programming course [28], especially because very detailed data can be gathered and analysed, including specific data during programming, for example line-level edits [29].

III. RESEARCH

At the University of Zagreb Faculty of Organization and Informatics (UNIZG FOI), from the academic year 2022/2023, the Introduction to Formal Methods course is offered as a revised Introduction to Knowledge Modeling course. The course redesign was also based on the results of the research that included analysing a five-year data (academic years 2016/2017-2020/2021) obtained from student surveys and their course performance, including final grades. The goal of the whole research was to improve the quality of teaching and learning process. The part of the research included comparison of students' perception of their understanding of the content of the course compared to the actual understanding that is represented with their final grades.

At the end of each execution of the course students completed surveys that collected information on

TABLE I. ANNUAL NUMBER OF ENROLLED AND SURVEYED STUDENTS

Year	No. of enrolled students	No. of completed surveys	Q 5	Q 6	Q 11	Q 14	% of completed surveys
2016	182	154	154	151	154	*	84.6
2017	205	177	175	177	177	176	86.3
2018	159	126	126	124	124	126	79.2
2019	165	144	144	144	144	142	87.3
2020	153	74	74	74	74	74	48.4
Total	**864**	**675**	**673**	**670**	**673**	**518**	**78.1**

understanding the course content, learning preferences, problems in understanding the Prolog, as well as their comments and recommendations. In Table 1 can be seen that out of 864 enrolled students during 5 years, 675 completed the survey. After online classes during COVID-19 pandemic in academic year 2020/2021 there was a lower response to the survey. Number of responses to questions that show students' perception about understanding course contents in general and of the practical part (Q5, Q6, Q11 and Q14) are also presented in the table. In Question 5 students were asked to asses their understanding of the course content in general and in Question 6 to assess their understanding of the practical part (exercises on computers); for both questions possible answers were low, medium and high. In Question 11 students were asked to asses the ammount of learning material for Prolog as insufficient, optimal or too extensive, and in Question 14 to asses whether team assignment helped them in learning Prolog (yes or no).

Learning outcomes adoption during each semester was tested with several assesment elements that contributed to final grade in the following percentages: two midterm exams - 60% (each 30%), seminar paper - 30%, activity - 5%, and seminar attendance - 5%. Direct knowledge of Prolog programming was assesed during midterm exams with 45% of the exams points (27% of the grade), knowledge of resolution reasoning procedure that Prolog uses for reasoning with 31.7% of the exams points (19% of the grade) and theoretical knowledge of logic needed for understanding practical part with 23.3% of the exams points (14% of the grade). Therefore, testing the practical knowledge of logic programming (Prolog and resolution) directly contributed to the final grade with 46%. In Table 2 are presented grades that students received after course completion. For comparison with survey questions about perception, grades are represented as low (grade 1), medium (grades 2 and 3) and high (grades 4 and 5) course understanding. Each point is 1% of the grade.

TABLE II. ANNUAL NUMBER OF ENROLLED AND PASSED STUDENTS

Year	Number of enrolled students	Percentage of grades (1; 0-49 points)	Percentage of grades (2 or 3; 50-75 points)	Percentage of grades (4 or 5; 76-100 points)
2016	182	31	67	2
2017	205	47	52	1
2018	159	41	54	5
2019	165	42	54	4
2020	153	53	47	0
Total	**864**	**43**	**55**	**2**

IV. RESULTS

A. *Student Perception of the Understanding of the Entire Course and Exercises*

Results were obtained using simple descriptive analytics of student surveys and grades. Questions 5 and 6 focus on student perception how well they understand the course content in general and additionally practical part – exercises on computers. Fig. 2 shows the results of the perception of understanding the entire course content (the theoretical knowledge of logic, knowledge of reasoning procedure and knowledge of logic programming). It is evident that the students to the greatest extent (73%-81%) perceive an average understanding of the entire course content. The remaining students perceive their understanding of the entire course as low (9%-18%) and high (7%-11%) throughout the observed years. The overall perception of the entire course content presented in Fig. 3 shows that on average 77% of students perceive their understanding of the entire course as a medium, 14% as low, and 9% as high.

Similar results were obtained when perceiving understanding of the exercises on computers (logic programming in Prolog) was observed. According to Fig. 4 and 5, students still perceive the understanding of the content of the exercises as medium (67% in total or 57%-71%, depending on the observed year). The only difference (in terms of the perception of the overall content) is that more students perceive the understanding of the exercises as high (22% overall, i.e. 9%-31% over the years), and slightly fewer students perceive it as low (11% overall, 3% -22% throughout the observed years).

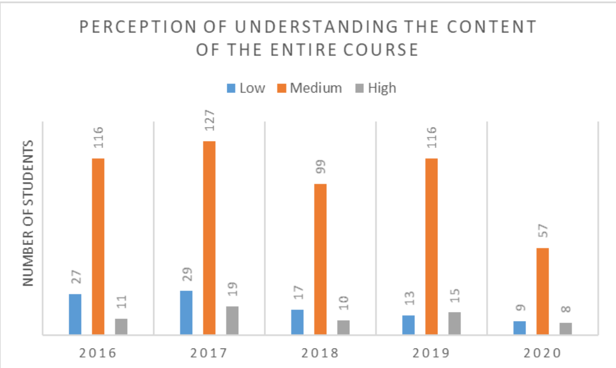

Figure 2. Perception of understanding the content of the entire course (2016-2020; N = 673)

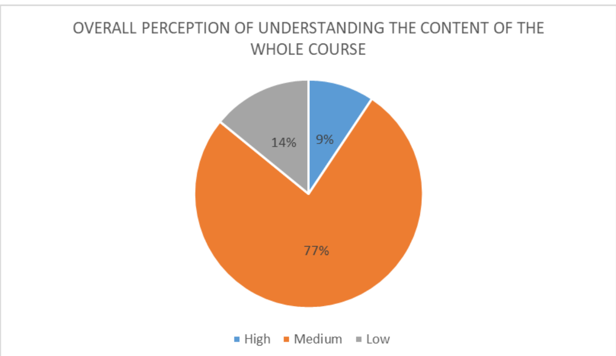

Figure 3. The overall perception of understanding the content of the entire course (2016-2020; N = 673)

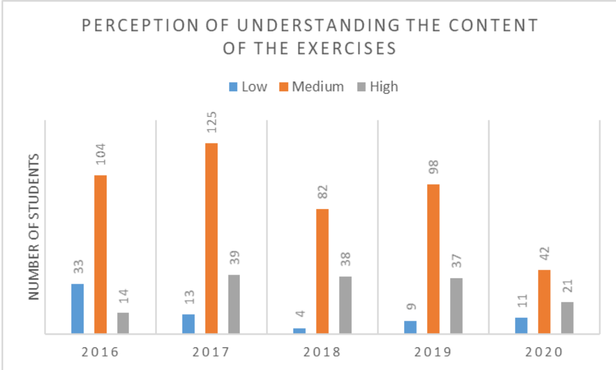

Figure 4. Perception of understanding the content of the exercises (2016-2020; N = 670)

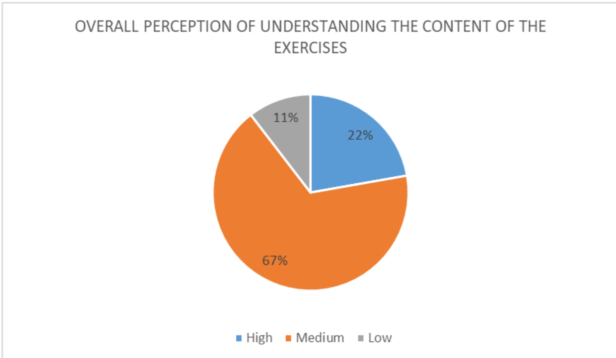

Figure 5. The overall perception of understanding the content of the exercises (2016-2020; N = 670)

Comparison of students' perceptions of understanding the course content (entire or only exercises) and final grades (Fig. 6. and 7.) indicates considerable differences between the perception and the real understanding of the content. According to the analysed data, 43% of all students had a low understanding of the course content (they achieved 0-49 points at assesment elements during semester and did not successfully pass the course). The data show that a majority of all students, 55%, showed a medium understanding of the course content (between 50 and 75 achieved points). The actual percentage is admittedly lower than the perceived one (concerning the content of the entire course and the exercises). However, it is still the closest to perception. Moreover, only 2% of all students (as opposed to 9% and 22% according to self-perception) showed high understanding of the course content (according to 76-100 achieved points).

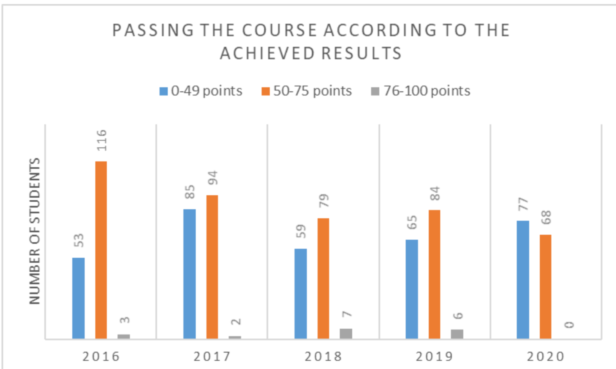

Figure 6. Passing the course according to the achieved results (2016-2020; N = 798)

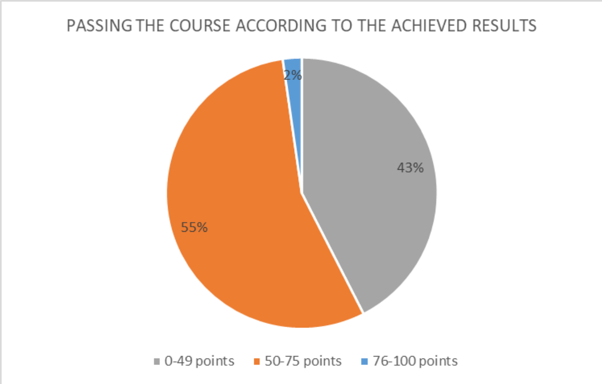

Figure 7. Passing the course according to the overall achieved results (2016-2020; N = 798)

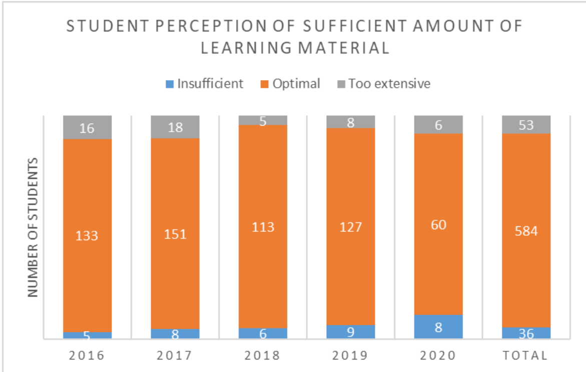

Figure 8. Student perception of sufficient amount of learning material (2016-2020; N = 673)

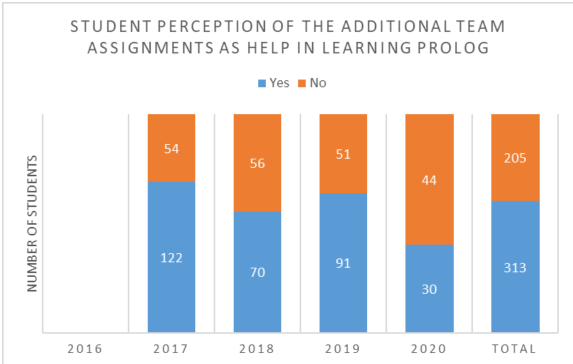

Figure 9. Student perception of the additional team assignments as help in learning Prolog (2016-2020; N = 518)

Previous research with other survey information about bigest problems in mastering the Prolog (syntax, logic/semantics, reasoning procedure, declarative paradigm, and examples used in class) identified declarative paradigm as the biggest problem and other elements as less problematic [30], which additionally contributes to student perception that they understand practical part of the course at least at the medium level.

B. Student Perception of the Learning Materials and Team Assignments

Additional information about student perception of their understanding of the course can be obtained with Questions 11 and 14, since practical knowledge of logic programming (Prolog and resolution) directly contributed to the final grade with 46%.

Answers to Question 11 show whether students think that the ammount of learning material available is optimal for learning, thus indirectly showing how they perceive their understanding. Most students over years answered that the ammount was optimal (as shown in Fig. 8). In total, 87% of students believe that the material that includes exercises for learning the basics of Prolog is optimal, 8% that it is too extensive and 5% that it is insufficient. When compared to grades in Table 2, even bigger difference can be noticed than with Questions 5 and 6. Although it must be taken into account that this is indirect indicator with smaller significance, the difference is too big to be dismissed.

Students were given team assignments from academic year 2017/2018, because survey from previous year showed that they think that it would help them to better learn Prolog. This is why results for Question 14 in Fig. 9. that explore how students perceive the role of team assignment in understanding Prolog are not given for academic year 2016/2017. Possible answers were yes and no, so they can not be compared in the same way with grades as previous questions. Additionally, givent that results show that on average little more than half of students - 60% - think that the team assignment helped them in understanding Prolog and that practical knowledge of logic programming directly contributed to the final grade with 46%, results from this question can not give valid information whether students perceive their understanding of the course content correctly.

The paralel with the previous research [30] mentioned in subsection IV.A can be applied also here. Results of this research about problems in learning Prolog [30] show that students think that the language itself does not present a problem, although they consider declarative pardigm as one. Most of the students do not perceive practical application of Prolog, which decreasses motivation and consequently understanding, as well as final grade [30].

V. Conclusion

Following the research, the complexity of the details that need to be taken into account and the importance of the students' opinion can be seen. In the previous research, the contemporary importance of Prolog and the paradox of its unpopularity was pointed out [30]. Also, problems in learning its basics through theory and practice were analysed and the effort invested in motivating students, creating exciting materials, and challenges in teaching methodology and achieving results within the online environment forced by COVID-19 pandemic were described [30].

To more understand the problems students face when learning logic programming, in this research student perception of the understanding of the course content was analyzed and then compared with the grades at the end of the semester. Results of the comparison showed that students perceive their understanding of the course content higher than the grades can confirm. Only 14% of students perceived their understanding of the entire course content as low, and most of them (87%) thought that the ammount of material is optimal, but 43% of them didn't pass the course after assesment during semester. In this case only final grades were taken into consideration, but many other elements of programming can be analysed and the difference in perception of understanding specific elements and real understanding can give valuable information, if observed throughout semester to enable adjustment of teaching pace and other teaching elements.

It is also necessary to refer to the shortcomings of the research. In the first observed year there was only one teaching assistant (later were two), which can possibly explain less negative grades in this year. In the first year there was also no team assignment, which can affect the results to some extent. In the last observed year teaching was conducted only online, resulting in significantly less completed surveys, again affecting the result to some extent. It is interesting that in this last year answers to questions were relatively similar, but it was the only year in which there were more negative than positive grades.

Future research will include analysis of other student engagements throughout semester and comparison with specific assesment results and assesment timing. Analysis of student perception of their understanding will be performed throughout semester and compared to results of assesements with the goal to optimize student learning processs and course performance.

ACKNOWLEDGMENT

This work has been fully supported by the Croatian Science Foundation under the project IP-2020-02-5071.

REFERENCES

[1] V. Lifschitz, “Foundations of logic programming,” in Principles of knowledge representation, G. Brewka, Ed. Stanford: CSLI Publications, 1996, pp. 69-127.

[2] M. R. Genesereth, and M. L. Ginsberg, “Logic programming,” Communications of the ACM, vol. 28, no. 9, pp. 933-941, September 1985.

[3] V. Aleksić, and M. Ivanović, “Introductory programming subject in European higher education,” Informatics in Education, vol. 15, no. 2, pp. 163-182, October 2016.

[4] P. Körner et al. “Fifty years of Prolog and beyond,” Theory and Practice of Logic Programming, vol. 22, no. 6, pp. 776-858, November 2022.

[5] A. Colmerauer, and P. Roussel, “The birth of Prolog,” in History of programming languages II, T. J. Bergin and R. G. Gibson, Eds. Boston: Addison-Wesley Professional, 1996, pp. 331-367.

[6] P. Brna et al., “Prolog programming techniques,” Instructional Science, vol. 20, no. 2-3, pp. 111–133, March 1991.

[7] I. Bratko, “Prolog Programming for Artificial Intelligence,” 4th Edition, North York: Pearson Education, 2012.

[8] W. Ertel, “Introduction to artificial intelligence,” Cham: Springer, 2018.

[9] F. Fei, J. Xie, and Z. B. Zhong, “Analysis on the application of intelligent robots based on Prolog language in primary and secondary education in the era of digital intelligence,” in Proc. of 3rd International Conference on Education, Knowledge and Information Management (ICEKIM), pp. 305-309, January 2022.

[10] R. Buscaroli et al. “A Prolog application for reasoning on maths puzzles with diagrams,” Journal of Experimental & Theoretical Artificial Intelligence, in press (published online 19 April 2022)

[11] A. Lally, and P. Fodor, “Natural Language Processing with Prolog in the IBM Watson System,” The Association for Logic Programming (ALP) Newsletter, 9, 2011.

[12] D. Radošević, T. Orehovački, and A. Lovrenčić, “New approaches and tools in teaching programming,” in Proc. of Central European Conference on Information and Intelligent Systems, pp. 49-57, September 2009.

[13] C. S. Cheah, “Factors contributing to the difficulties in teaching and learning of computer programming: A literature review,” Contemporary Educational Technology, vol. 12, no.2, ep272, October 2020.

[14] F. Demir, “The effect of different usage of the educational programming language in programming education on the programming anxiety and achievement,” Education and Information Technologies, vol. 27, no. 3, pp. 4171-4194, April 2022.

[15] T. Jenkins, „On the difficulty of learning to program,” in Proc. of the 3rd Annual LTSN-ICS Conference, pp. 53-58, 2002.

[16] A. Pears et al., “A survey of literature on the teaching of introductory programming,” ACM SIGCSE Bulletin, vol. 39, no. 4, pp. 204-223, December 2007.

[17] D. Callear, “Teaching Programming: Some Lessons from Prolog,” in proc. 8th of Annual Conference on Teaching and Learning, 2000.

[18] S. A. Rebelsky et al., “Why I do declare! declarative programming in the undergraduate curriculum,” in Proc. of SIGCSE technical symposium on Computer Science Education, pp. 398-399, February 2001.

[19] S. Yang, and M. Joy, “Approaches for Learning Prolog Programming,” Innovation in Teaching and Learning in Information and Computer Sciences, vol. 6, no. 4, pp. 88-107, 2007.

[20] A. Stathaki et al., “i-Prolog: a web-based intelligent tutoring system for learning Prolog,” in Interactivity, Game Creation, Design, Learning, and Innovation, A. L. Brooks, E. Brooks and N. Vidakis, Eds., Cham: Springer 2017, pp. 337-346.

[21] G. Siemens, “Learning analytics: The emergence of a discipline,” American Behavioral Scientist, vol. 57, no. 10, pp. 1380-1400, October 2013.

[22] D. Clow, “An overview of learning analytics,” Teaching in Higher Education, vol. 18, no. 6, pp. 683-695, 2013.

[23] O. Viberg, M. Hatakka, O. Bälter and A. Mavroudi. “The current landscape of learning analytics in higher education,” Computers in human behavior, vol. 89, pp. 98-110, December 2018.

[24] M. Hernández-de-Menéndez et al., “Learning analytics: state of the art,” International Journal on Interactive Design and Manufacturing, vol. 16, no.3 pp. 1209-1230, September 2022.

[25] U. Omer, et al., “Learning analytics in programming courses: Review and implications,” Education and Information Technologies, in press (published online 16 February 2023)

[26] M. Berland, T. Martin, T. Benton, C. P. Smith, and D. Davis, “Using learning analytics to understand the learning pathways of novice programmers,” Journal of the Learning Sciences, vol. 22, no. 4, pp. 564-599, October-December 2013.

[27] P. Blikstein, M. Worsley, C. Piech, M. Sahami, S. Cooper and D. Koller, “Programming pluralism: Using learning analytics to detect patterns in the learning of computer programming,” Journal of the Learning Sciences, vol. 23, no. 4, pp. 561-599, October-December 2014.

[28] D. Azcona, I-H. Hsiao and A. F. Smeaton, “Detecting students-at-risk in computer programming classes with learning analytics from students’ digital footprints,” User Modeling and User-Adapted Interaction, vol. 29, no. 4, pp. 759-788, September 2019.

[29] P. Ihanthola et al., “Educational Data Mining and Learning Analytics in Programming: Literature Review and Case Studies,” in Proc. of 2015 ITiCSE on Working Group Reports, pp. 41-63, 2015.

[30] V. Sekovanić and S. Lovrenčić, “Challenges in teaching logic programming,” in Proc. of the 45th International Convention MIPRO 2022, pp. 658-662, May 2022.