Commitment Threshold – On Student Retention in MOOCs

Erkki Kaila
University of Turku
Turku, Finland
University of Helsinki
Helsinki, Finland

Antti Laaksonen
University of Helsinki
Helsinki, Finland

Emilia Oikarinen
University of Helsinki
Helsinki, Finland

Kjell Lemström
University of Helsinki
Helsinki, Finland

ABSTRACT

Learning and teaching at scale became especially topical and important matters due to the COVID-19 pandemic. The obvious way to that end is to use Massive Open Online Courses (MOOCs). However, the pedagogics and student behaviour between online and classroom implementations are far from similar, and successful implementation of a MOOC requires an understanding of this. Typically, the dropout rates of MOOCs are pretty high. Most of the dropouts happen in the very early stages of the course. This paper studies how and when students commit to MOOC courses. For this study, we have selected four courses with different implementations from our extensive portfolio of MOOC courses at University of Helsinki. Based on the results, the students who complete the first two parts (or weeks) of a more prolonged course are usually quite committed to completing the entire course. The behaviour varies slightly between scheduled and unscheduled courses and between courses of different topics. The inner and outer motivations also play a role in commitment. Regardless, it seems that the authors of MOOCs should pay special attention to making the first steps as engaging as possible.

CCS CONCEPTS

• **Social and professional topics → Computing education**.

KEYWORDS

MOOCs, Student Retention, Commitment Threshold

ACM Reference Format:
Erkki Kaila, Antti Laaksonen, Emilia Oikarinen, and Kjell Lemström. 2023. Commitment Threshold – On Student Retention in MOOCs. In *International Conference on Computer Systems and Technologies 2023 (CompSysTech '23), June 16–17, 2023, Ruse, Bulgaria.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3606305.3606322

1 INTRODUCTION

A typical phenomenon (see, e.g. [6], [19]) with Massive Open Online Courses, or MOOCs, is that many students drop out in the very early phases. It is assumed that many students are just sitting in by curiosity without any real intention of completing the course, and there is little to be done to retain them on the course. On the other hand, the commitment can arise rather quickly with some thoughtful arrangements and built-in hooks in the course material.

In this paper, we study the student commitment in various MOOCs provided by University of Helsinki, with a focus on how the retention rate behaves as a function of elapsed time. If there is a significant threshold in the retention rate, or a *commitment threshold*, at some point, knowledge of such a phenomenon could be a valuable tool. Given that the resources are always limited, the input/output ratio for any severe actions to retain the students on the course would be the highest from this point onward. In our experience, it is essential to lure the students to start the MOOC with some exciting content, and forcing them to register right at the beginning would be quite the opposite. Moreover, coming from a small language region, as we do in University of Helsinki, we have found it important to offer the MOOCs in the local language to lower the first hurdle for starting the courses.

If we found such a commitment threshold, the ideal moment for the course registration could be just after that. This way, we could maximize the study credit accumulation by minimizing the number of students forgetting to register and apply for the credits.

2 RELATED WORK

MOOCs (see, e.g. [11], [22]) have permanently changed education. As stated in [16], "MOOC generally carries no fees, no prerequisites other than Internet access and interest, no predefined expectations for participation, and no formal accreditation". Over time, several MOOCs have broken some of those principles (for example, prerequisites [20], participation [7] and accreditation [3]). However, the general idea of free education available for everyone still prevails.

For programming MOOCs, some key features are automatic assessment of student solutions ([2], [12], [18]) and immediate and automated feedback (see, e.g., [15], [5]). These enable students to take the role of active learners regardless of time and space, which is generally seen highly important in learning to program. For example, in [21] the authors state that requiring participants to be more active can lead to excellent results. According to [1], interactivity plays a role in formative assessment in addition to activating and engaging students.

Traditionally, MOOCs are considered to have high drop-out rates. For example, the completion rate of a MOOC can be as low as 5% [6].

CompSysTech '23, June 16–17, 2023, Ruse, Bulgaria

ACM ISBN 979-8-4007-0047-7/23/06.
https://doi.org/10.1145/3606305.3606322

The authors of [4] introduced a machine learning based tool for predicting student drop-outs in data structures and algorithms courses. There are various other similar approaches reported (see, e.g., [8], [13], [23]), which seems to underline the fact that the dropouts are a real problem with MOOCs. In their literature review, the authors of [10] list lack of time and motivation, feeling of isolation and lack of interactivity as the biggest reasons for student drop-outs in MOOCs. Moreover, [24] state that "researchers lack a solid understanding of what student needs are being addressed by MOOCs, and how well MOOCs now address these needs".

In [19], the authors observed that the drop-out rate in programming MOOCs is highest at the beginning of the course. They also found that the students succeeded well in easier courses until the drop-out, but in more difficult ones, the drop-out seemed to be related to a decrease in course performance. The authors of [14] list factors that are rated higher among participants who complete MOOCs, including "personal sustainability of distance learning", "social influence" and "usefulness related to certification".

3 METHODOLOGY

3.1 MOOC Instances

We observed 4 different MOOCs and a total of 7 instances for this study. Table 1 lists the course instances and the associated acronyms. The table also lists the number of students (N), the number of students who passed the course (N'), the fraction of students passing the course (C), and the percentage of points needed to complete each part (δ).

For programming MOOC, two very different instances were selected. The SP instance was organized in the spring of 2020. The course was taught using Java as the programming language. The course lasted for 14 weeks, and for each week, we opened one part (or section) of the course. After the course, students had three possibilities to take the course exam. As a unique feature, the course acted as an entrance exam to University of Helsinki. Students who completed the course well enough could take a separate entrance exam, and the best students were admitted into the university. The UP instance was organized in 2021. In this instance, we changed the programming language to Python. In this instance, the students had no hard deadlines (except the one at the end of the year). Instead, all parts were available throughout the year. There were nine possibilities to take the course exams. The minimum limit required from each round was also lower in the UP instance, only 25 percent of the points was needed from each part (in comparison to 80 % in the SP instance). Each subsequent part was opened when the student met the minimum requirement in the previous part.

Table 1: Basic statistics of the MOOC courses discussed. N, N', C and δ denote the number of students, the number of students who passed the course, the fraction of students who passed the course ($C = N'/N$) and the threshold of solved assignments to pass an individual part.

Course	Acronym	N	N'	C	δ
Scheduled Programming	SP	3972	1256	0.32	0.8
Unscheduled Programming	UP	9998	1141	0.11	0.25
Data Analysis 2020	DA20	1873	267	0.14	0.8
Data Analysis 2021	DA21	751	128	0.17	0.8
Data Structures and Algorithms 2020	DSA20	408	235	0.58	0.625
Data Structures and Algorithms 2021	DSA21	307	161	0.52	0.625
Introduction to Databases	ID	1404	660	0.47	0.6

For the DA MOOC, we observed two instances with identical setups. DA20 and DA21 were organized between June 2020 and April 2021, and May 2021 and October 2021, respectively. The course uses Python as the programming language for data analysis tasks. The MOOC material contains six parts of programming assignments, and similarly to the UP MOOC, there were no hard deadlines for the individual parts. The students could proceed to the next part by completing 80% of the previous part. In addition to automatically assessed exercises (the MOOC part), the DA course contained a peer-reviewed project work and exam. The students could choose their schedule, as there were four exam/project work deadlines in the 2020 and three deadlines in the 2021 course instances. A fraction of students chose only to complete the automatically assessed programming exercises (parts 1–6) and not to do the exam or the project work: in the 2020 instance 373 students and in the 2021 instance, 173 students completed parts 1–6, in comparison to 267 and 128 students, respectively, who completed the entire course (see Table 1).

The DSA MOOC consists of seven parts. We observed two course instances, DSA20 and DSA21, organized in fall 2020 and 2021, respectively. Both instances were organized using the same format and material. The students solved programming and other exercises with automatic evaluation in each part. Each part contained eight exercises, and it was required to solve at least five exercises to advance to the next part and complete the course. Thus, a student who completed the course could solve between 35 and 56 exercises. In 2020, 408 students started the course, and 235 of them completed the course. In 2021, there were 307 students, and 161 completed the course. A probable reason for the smaller number of students in 2021 was a degree transition which forced many students to complete the course in 2020.

The ID MOOC contains a collection of 100 SQL exercises. In each exercise, the task is to design a SQL query corresponding to the problem statement. It is required to solve at least 60 exercises to complete the course. The exercises can be solved in any order, and all the exercises have the same deadline. We considered here the exercises solved during the 2021 version of the course. Since the exercises can be solved in any order, we only focused on each student's total number of exercises.

3.2 Implementation of MOOCs

The assignments in the MOOCs were implemented with the Test My Code (TMC) environment [17] enabling automatic assessment and immediate feedback. The exercises were embedded within learning material consisting of text, images, examples, short slide shows, and animations. In the first three weeks of the UP MOOC, the students could complete all the programming tasks in the browser with no extra tools or plugins required. After the first three weeks in the UP MOOC (and since the beginning in the other MOOCs), the assignments were completed using an external editor. This was to provide a real-world experience in completing the tasks.

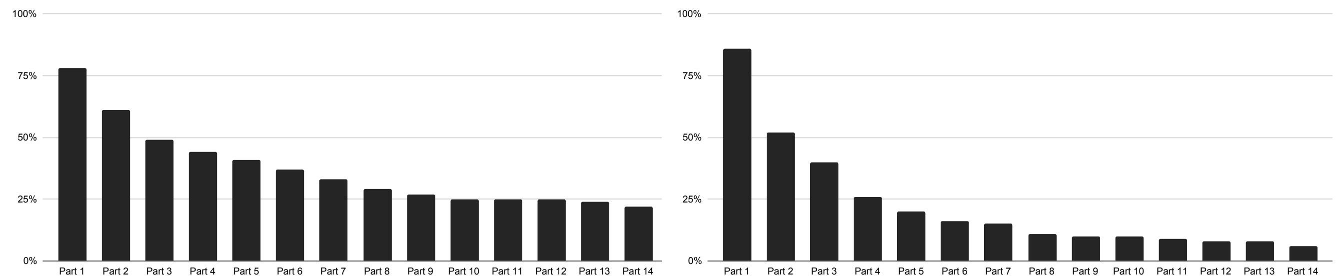

Figure 1: The completion rates of all 14 parts of the SP MOOC (left) and the UP MOOC (right). The minimum requirement for completing a part in the SP MOOC was 80% of available points, whereas it was 25% of the UP MOOC.

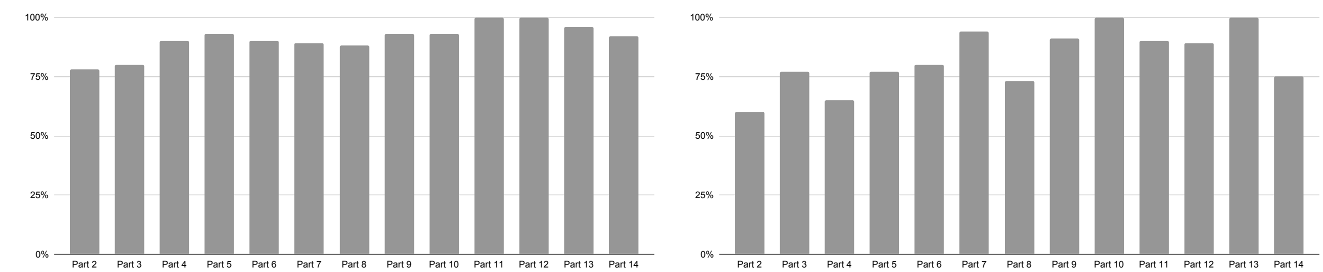

Figure 2: The relative retention rates of the final 13 parts of the SP MOOC (left) and the UP MOOC (right).

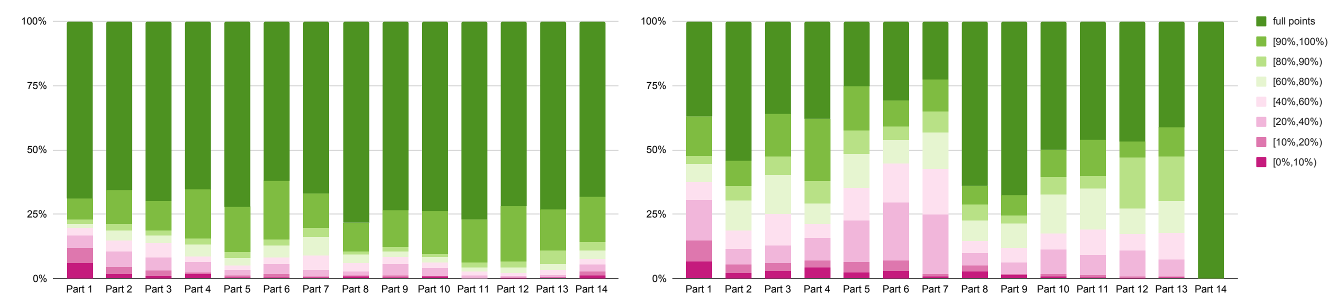

Figure 3: Percentages of students achieving a certain fraction of points on each part of programming SP (left) and UP (right) MOOCs. The intervals in legend are open at the upper limit, i.e., $y \in [x_1, x_2)$ denotes $x_1 \leq y < x_2$ for a fraction of points y.

4 RESULTS

In this section, we present the results of our study.

4.1 Programming MOOCs

Fig. 1 (left) displays the completion rates of all parts in the SP MOOC. Clearly, the most significant drop-down rates occur between parts 1 and 2 and parts 2 and 3. After part 3, the students seem to be more tightly committed to the course. This can be verified by observing the *relative retention rate* of each part (the number of students who continued from the previous part) displayed in Fig. 2(left).

A similar figure displaying the pass rates in the UP MOOC is shown in Fig. 1(right). Again, the most significant drop-down rate occurs between the first two parts. However, in the UP instance, the commitment does not seem as tight as in the SP instance until part 5. The relative retention rates for all parts are displayed in Fig. 2(right). It can be seen that the retention rate grows towards the end of the MOOC, but not as steadily as in the UP instance.

We further studied the fractions of completed exercises for the SP and UP instances of the programming MOOC. Fig. 3 shows the fraction of students completing a certain percentage of points available on each part. For each part, we only consider students who

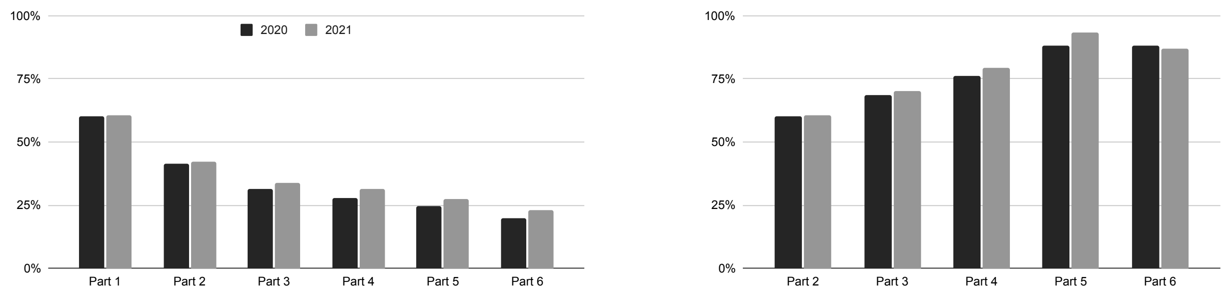

Figure 4: Left: Completion rates of DA MOOCS. The minimum requirement for completing a part in the MOOCS was 80% of available points. Right: Relative retention rates of DA MOOC's last 5 parts.

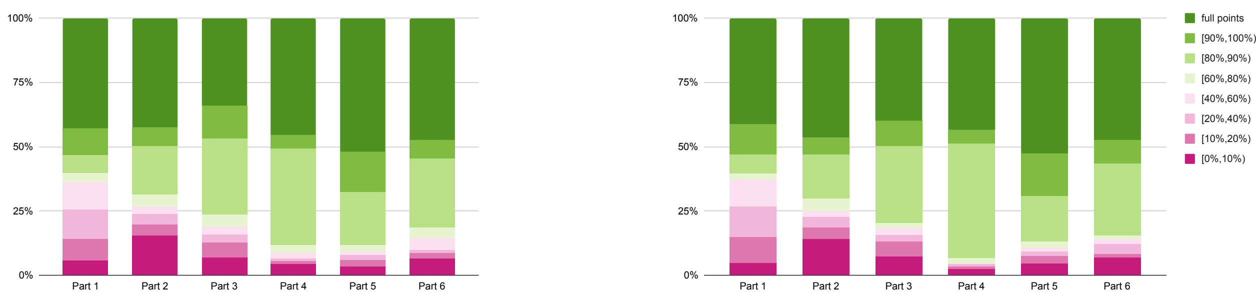

Figure 5: Percentages of students achieving certain fraction of points on each part of DA20 (left) and DA21 (right) instances. The intervals listed in legend are open at the upper limit, i.e., we use $y \in [x_1, x_2)$ to denote $x_1 \leq y < x_2$ for a fraction of points y.

have completed all previous parts, and initially we consider only students who completed at least one point. The higher requirement for passing each part (80% versus 25%) in the scheduled course is clearly visible. Still, it should be noted that in both instances, majority of students complete more than required number of exercises in each part. In the SP course, majority of students completed all exercises in each part (even though there is no clear extrinsic motivation for this). A similar phenomenon on a lesser scale is visible in the UP course as well. The full completion of part 14 in the UP instance can be explained by the fact that only one point was available as the final part consisted of writing your own game.

4.2 Data Analysis MOOC

Fig. 4(left) displays the completion rates of all parts in DA MOOC instances for 2020 and 2021. There is a significant drop in the number of students in the first two parts, but after this the commitment is tight. The behavior of the students in the two instances is very similar: students in 2021 are slightly more committed, but the difference is small and likely not significant. The similarity is expected because the instances are identical in terms of their organization, materials, and exercises. Fig. 4(right) displays the relative retention rate of each part, and seems to confirm the observation.

We further analyzed the fractions of completed exercises for each part, similarly to Fig. 3. For DA MOOCs 80% of the available points are required to pass a part and move on to the next. There is no external reward for achieving more than 80% of the points on each part (course grading is fully based on grading of the exam and project work). Fig. 5 shows the fraction of students completing a certain percentage available on each part (for each part, we only consider students who have completed all previous parts). Clearly, it is quite rare that students drop out if they have completed more than 20% of the exercises (excluding part 1). Also, a surprisingly high fraction completes 100% of exercises without any external reward.

4.3 Data Structures and Algorithms MOOC

Fig. 6(left) shows the portion of students who completed each part in 2020 and 2021. Fig. 6(right) shows the retention rate for each part of the DSA MOOC. It can be seen that the number of students remains relatively stable after the second part. Thus, it seems that a student typically either completes all parts of the course or drops out before the third part of the course.

Fig. 7 shows the distribution of points among the students who completed the course. Since it is required to solve between 5 and 8 exercises in each part, the total number of points to pass is between 35 and 56. It can be seen that there are five peaks at 36, 40, 44, 48 and 52 points. These peaks correspond to course grades between 1 and 5, respectively. In addition, there is a peak at 56 points, which shows that some students want to solve all exercises even if it is not required for the highest grade.

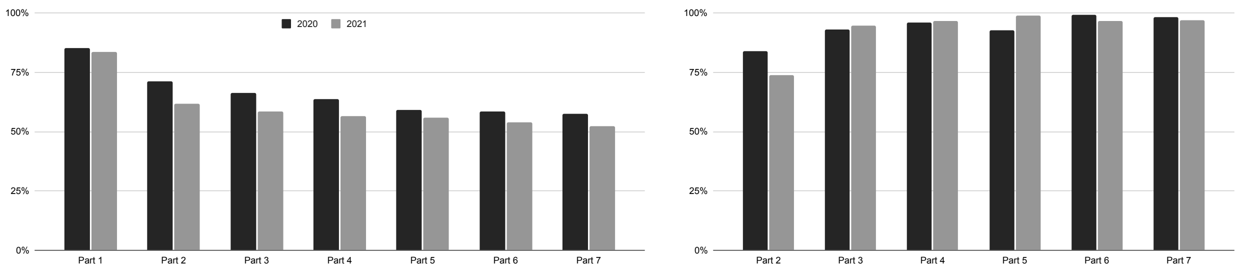

Figure 6: Left: Completion rates of the DSA MOOC. The minimum requirement for completing a part in the MOOC was 5/8 of available points. Right: Relative retention rates of the last six parts of the DSA MOOC.

4.4 Introduction to Databases MOOC

Fig. 8 shows the distribution of solved exercises in 2021 in the ID MOOC. Four peaks can be seen at 20, 60, 90 and 100 points. The first peak can be explained by the fact that the first 20 exercises deal with topics discussed in the first chapter of the course material. The peaks at 60 and 90 points correspond to the required number of exercises to pass the course and get the highest grade. Finally, like in other courses, the last peak at 100 points shows that some students want to solve all course exercises, even if this would not be needed to get the highest grade.

This course differs from the other courses considered in this paper because it does not consist of individual parts (or sections). Instead, all exercises are available when the course begins and can be solved in any order. There is no clear point where many students would drop out. On the other hand, the number of students who solve between 40 and 59 points is small. If students can solve at least 40 exercises, they finally solve enough exercises to pass the course.

5 DISCUSSION

This paper studies students' behaviour in several computer science MOOC course instances with somewhat different arrangements. As suspected, we found that a significant proportion of dropouts occurs within the first two weeks (or parts) independently of the arrangements (see, e.g. Fig. 4(left), where the partly completion rate becomes much steadier in the third part). Thus, it seems that the commitment threshold takes place at the beginning of the third section (or part) of a MOOC course. This finding is more substantial than (though on a par with) [9], where it is noted that the students are likely to complete a MOOC after the middle point. It should be noted that even though some courses require extra engagement (for example, the programming MOOC involves the installation of an external programming editor after week 3) later, the commitment seems to hold after the first three weeks.

Examining the completion rate distributions on weekly or partial assignments, it is not surprising that there is a peak at the minimum required level. We could find other extrinsic motivator effects in grading borders: in Fig. 7, this effect appears strikingly in the DSA course. However, finding out that so many students go far beyond that level was spontaneous. For instance, in the UP MOOC, a significant proportion of retained students do over 60 % of the partial assignments even though the minimum required level is set to 25 % (SP MOOC's accumulation of points is, in fact, higher, but the phenomenon does not manifest itself as flashy since the minimum required level is set so high, to 80 %). Furthermore, in all our MOOC instances, a surprisingly remarkable share of students do all the assignments without getting an extra reward from such extrinsic travail (see, e.g., Fig. 8 where one of the highest peaks in the number of solved exercises appears at 100/100 spot). In future, we are interested in studying whether these very active students perform better in the courses and their studies in general.

Despite the apparent limitations of our setting, the results we found manifest themselves already in the statistics presented above. In the future, we will study these findings more systematically and controlled to find out the particular variables involved and whether some confounding or interacting factors are taking place. Moreover, we are interested in studying how the students' commitment carries over to the following course(s) once they complete a MOOC.

6 CONCLUSIONS

We studied four MOOC courses at the University of Helsinki with somewhat different implementations. We aimed to find whether we could see similarities in the student course commitment rates as a function of time. We were especially eager to discover whether we could find a time point or a commitment threshold, after which students retained that are likely to complete the whole course. In the three courses that had an arrangement that supported this phenomenon to be studied, such a time point appears approximately at the beginning of the third week/part of the course. Hence, it seems the students are fully committed to the course's third part (or section). We also found, unsurprisingly, that the effects of the extrinsic course motivators (minimum requirement level of exercises and the grading borders) can be seen as peaks in the achieved points distribution. However, we were surprised to find that many students go far beyond the minimum passing requirement level of assignments and that many students are eager to complete all the given assignments with no external reward.

REFERENCES

[1] Carlos Alario-Hoyos, C Delgado Kloos, Iria Estévez-Ayres, Carmen Fernández-Panadero, Jorge Blasco, Sergio Pastrana, and J Villena-Román. 2016. Interactive activities: the key to learning programming with MOOCs. In *Proc. European Stakeholder Summit on Experiences and Best Practices in and Around MOOCs, EMOOCS.* Books on Demand, Graz, Austria, 319–328.

[2] Anis Bey, Patrick Jermann, and Pierre Dillenbourg. 2018. A comparison between two automatic assessment approaches for programming: An empirical study on

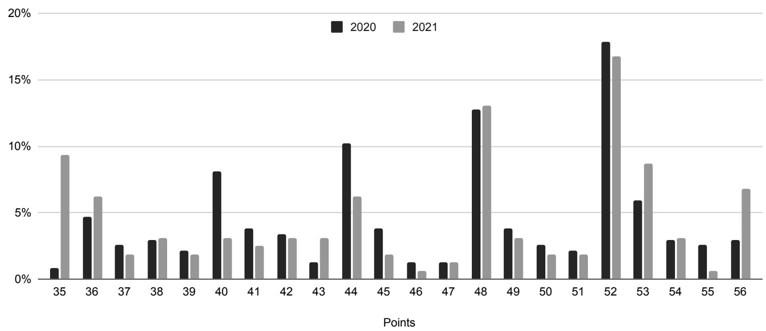

Figure 7: Distribution of points among the students who completed the DSA MOOC.

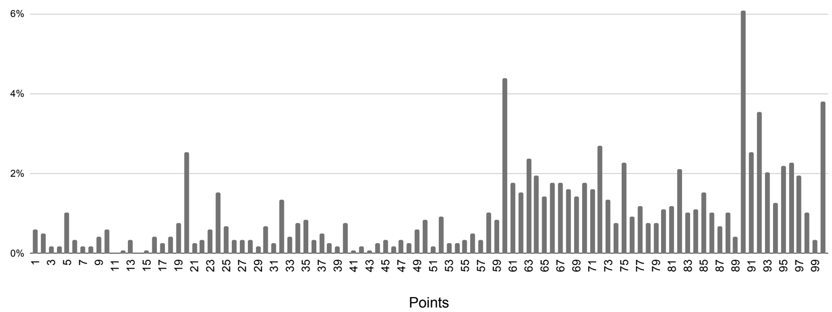

Figure 8: Distribution of points among the students who attended the ID MOOC.

MOOCs. *J. of Educational Technology & Society* 21, 2 (2018), 259–272. https://doi.org/10.30191/ETS.201804_21(2).0022

[3] Amit Chauhan. 2014. Massive open online courses (MOOCS): Emerging trends in assessment and accreditation. *Digital Education Review* 25 (2014), 7–17. https://doi.org/10.1344/der.2014.25.7-17

[4] Yunfan Chen and Ming Zhang. 2017. MOOC student dropout: Pattern and prevention. In *Proc. ACM Turing 50th Celebration Conference-China*. ACM, Shanghai, China, 1–6. https://doi.org/10.1145/3063955.3063959

[5] Zhenghao Chen, Andy Nguyen, Amory Schlender, and Jiquan Ngiam. 2017. Real-time programming exercise feedback in MOOCs. In *Proc. 10th International Conference on Educational Data Mining*. ERIC, Wuhan, China, 414–415.

[6] Wenzheng Feng, Jie Tang, and Tracy Xiao Liu. 2019. Understanding dropouts in MOOCs. In *Proc. AAAI Conference on Artificial Intelligence*, Vol. 33 (01). AAAI, Honolulu, Hawaii, USA, 517–524. https://doi.org/10.1609/aaai.v33i01.3301517

[7] Oriol Borrás Gené, Margarita Martínez Núñez, and Ángel Fidalgo Blanco. 2014. Gamification in MOOC: challenges, opportunities and proposals for advancing MOOC model. In *Proc. Second International Conference on Technological Ecosystems for Enhancing Multiculturality*. Association for Computing Machinery, Salamanca, Spain, 215–220. https://doi.org/10.1145/2669711.2669902

[8] Sherif Halawa, Daniel Greene, and John Mitchell. 2014. Dropout prediction in MOOCs using learner activity features. *Proc. second European MOOC stakeholder summit* 37, 1 (2014), 58–65.

[9] Kate S Hone and Ghada R El Said. 2016. Exploring the factors affecting MOOC retention: A survey study. *Computers & Education* 98 (2016), 157–168. https://doi.org/10.1016/j.compedu.2016.03.016

[10] Hanan Khalil and Martin Ebner. 2014. MOOCs completion rates and possible methods to improve retention — A literature review. In *EdMedia+ innovate learning*. Association for the Advancement of Computing in Education (AACE), Tampere, Finland, 1305–1313.

[11] Paul Kim. 2015. *Massive open online courses.* Taylor & Francis, New York, NY, USA. https://doi.org/10.4324/9781315848655

[12] Marina Lepp, Piret Luik, Tauno Palts, Kaspar Papli, Reelika Suviste, Merilin Säde, Kaspar Hollo, Vello Vaherpuu, and Eno Tõnisson. 2016. Self-and automated assessment in programming MOOCs. In *International Computer Assisted Assessment Conference*. Springer, Springer, Tallinn, Estonia, 72–85. https://doi.org/10.1007/978-3-319-57744-9_7

[13] Jiajun Liang, Chao Li, and Li Zheng. 2016. Machine learning application in MOOCs: Dropout prediction. In *2016 11th International Conference on Computer Science & Education (ICCSE)*. IEEE, Nagoya, Japan, 52–57. https://doi.org/10.1109/ICCSE.2016.7581554

[14] Piret Luik, Marina Lepp, Tauno Palts, Merilin Säde, Reelika Suviste, Eno Tõnisson, and Maria Gaiduk. 2018. Completion of programming MOOC or dropping out: Are there any differences in motivation. In *Proc. 17th European Conference on e-Learning ECEL*. Academic Conferences and Publishing International, Athens, Greece, 329–337.

[15] Victor J Marin, Tobin Pereira, Srinivas Sridharan, and Carlos R Rivero. 2017. Automated personalized feedback in introductory Java programming MOOCs. In *2017 IEEE 33rd Int. Conference on Data Engineering (ICDE)*. IEEE, San Diego, CA, USA, 1259–1270. https://doi.org/10.1109/ICDE.2017.169

[16] Alexander McAuley, Bonnie Stewart, George Siemens, and Dave Cormier. 2010. *The MOOC model for digital practice.* Technical Report. University of Prince Edward Island. 64 pages. http://www.elearnspace.org/Articles/MOOC_Final.pdf

[17] Martin Pärtel, Matti Luukkainen, Arto Vihavainen, and Thomas Vikberg. 2013. Test my code. *Int. J. of Technology Enhanced Learning* 5, 3–4 (2013), 271–283. https://doi.org/10.1504/IJTEL.2013.059495

[18] Vreda Pieterse. 2013. Automated Assessment of Programming Assignments. In *Proc. of the 3rd Computer Science Education Research Conference on Computer Science Education Research*. Open Universiteit, Heerlen, Arnhem, Netherlands, 45–56.

[19] Marili Rõõm, Marina Lepp, and Piret Luik. 2021. Dropout Time and Learners' Performance in Computer Programming MOOCs. *Education Sciences* 11, 10 (2021), 643. https://doi.org/10.3390/educsci11100643

[20] Bronwen J Swinnerton, Neil P Morris, Stephanie Hotchkiss, and James D Pickering. 2017. The integration of an anatomy massive open online course (MOOC) into a medical anatomy curriculum. *Anatomical sciences education* 10, 1 (2017), 53–67. https://doi.org/10.1002/ase.1625

[21] Arto Vihavainen, Matti Luukkainen, and Jaakko Kurhila. 2012. Multi-faceted support for MOOC in programming. In *Proc. 13th annual conference on Information technology education*. ACM, Calgary, Alberta, Canada, 171–176. https://doi.org/10.1145/2380552.2380603

[22] Jochen Wulf, Ivo Blohm, Jan Marco Leimeister, and Walter Brenner. 2014. Massive open online courses. *Business & Inf. Systems Engineering* 6, 2 (2014), 111–114. https://doi.org/10.1007/s12599-014-0313-9

[23] Wanli Xing and Dongping Du. 2019. Dropout prediction in MOOCs: Using deep learning for personalized intervention. *J. of Educational Computing Research* 57, 3 (2019), 547–570. https://doi.org/10.1177/073563311875701

[24] Saijing Zheng, Mary Beth Rosson, Patrick C Shih, and John M Carroll. 2015. Understanding student motivation, behaviors and perceptions in MOOCs. In *Proc. 18th ACM conf. on computer supported cooperative work & social computing*. ACM, Vancouver, BC, Canada, 1882–1895. https://doi.org/10.1145/2675133.2675217