



Exploring the Connections Between the Use of an Automated Feedback System and Learning Behavior in a MOOC for Programming

Hagit Gabbay^(✉)  and Anat Cohen 

School of Education, Tel Aviv University, Ramat Aviv, P.O.B 39040, 69978 Tel Aviv, Israel
hagitgabbay@mail.tau.ac.il, anatco@tauex.tau.ac.il

Abstract. Automated Testing and Feedback (ATF) systems are widely applied in programming courses, providing learners with immediate feedback and facilitating hands-on practice. When it comes to Massive Open Online Courses (MOOCs), where students often struggle and instructors' assistance is scarce, ATF appears to be particularly essential. However, the impact of ATF on learning in MOOCs for programming is understudied. This study explores the connections between ATF usage and learning behavior, addressing relevant measures of learning in MOOCs. We extracted data of learners' engagement with the course material, code-submissions and self-reported questionnaire in a Python programming MOOC with an ATF system embedded, to compile an overall and unique picture of learning behavior. Learners' response to feedback was determined by sequence analysis of code submission, identifying improved or feedback-ignored re-submissions. Clusters of learners with common learning behaviors were identified, and their response to feedback was compared. We believe that our findings, as well as the holistic approach we propose to investigate ATF impact, will contribute to research in this field and to effective integration of ATF systems to maximize learning experience in MOOCs for programming.

Keywords: Automated feedback · MOOCs for programming · Clustering · Learning analytics

1 Introduction and Related Work

1.1 Automated Testing and Feedback (ATF) Systems

Writing and executing code is the basis for learning a programming language and developing programming skills [36]. An accurate, detailed and timely feedback on the correctness and quality of the code may promote learning and increase practice effectiveness [33]. Large scale courses, however, make assessing the great volume of submissions and giving individual feedback nearly impossible [17]. Therefore, Automated Testing and Feedback systems (ATF) are often offered as a learning tool, providing immediate feedback and allowing unlimited resubmissions [22]. Recent reviews of literature reveal that ATF tools and systems are widely available, developed using different technologies and

methodologies [9, 22, 30]. Feedback may refer to syntax errors, the correctness of results or efficiency of the code [15, 36]. It may consist of only result correctness, or it might include a detailed explanation of the error or hints for solving it [22, 35]. In response to feedback, the learner is required to take two steps: decide whether to resubmit or waive, and to engage in an active practice of identifying and correcting the errors [29].

Behavioral characteristics of learners using the ATF system have been studied mainly through analyzing the programs submitted to the system and the feedback received. Learners' progress through code assignments, for example, was analyzed in [28] using cluster analyses based on variables harvested from ATF logs. Machine learning algorithms were applied on code solutions submitted for course assignments to identify attrition points and predict dropouts [37]. These and similar studies, however, did not analyse learning behavior in light of all course resources, including content consumption and solving non-code exercises.

Regarding affective measures, studies have suggested that the automated feedback enhances satisfaction and sense of learning [3, 4]. Learners perceive the automated feedback as enhancing learning and increasing motivation and engagement [30]. However, results concerning the system's impact on performance in the course, represented by scores of final exam or concluding assignment, were inconclusive (e.g. [6, 16]).

1.2 Massive Open Online Courses (MOOCs) and Learning Behavior Measures

Recent years have seen an increase in MOOCs in a variety of subjects. Learners in MOOC are usually diverse in their motivation for learning, as well as in their demographics and previous background [1]. Despite high enrollment rates, a high percentage of learners do not complete their learning due to variety of reasons including the lack of prior knowledge, struggling with course materials, and the need to self-regulate learning [38]. MOOCs, on the other hand, are not necessarily for credit and completing the course is not the ultimate goal [13]. Different measures should therefore be applied to evaluate learning outcomes and success in MOOCs [12, 23]. A common indicator of learning outcomes in MOOCs is learner's engagement, measured by [20, 23] as the degree of interaction with course materials, e.g. watching videos and attempt to solve exercises. Persistence is another common measure, defined by learner's determination to complete assignments and the achieved progress in study units [20]. Grades achieved on exercises and assignments determine the performance in the course [18].

Applying cluster analysis, researchers identified learning behavioral patterns and categorized learner by common patterns. In a key study [23] identified four major groups of MOOC learners: completers (learners who completed most assignments), auditors (completed few exercises but engaged in watching videos), disengaging (stopped participating after solving few exercises), and sampling (watched only few videos along the course). Similar studies proposed from three up to seven clusters, categorizing learners based on various sets of learning characteristics (e.g. [2, 21]). The most common variables used were the number of videos watched, in-video questions answered, exercises and assignments submitted, and social engagement such as activity discussion forums. In current research, we considered the suggested measures of learning behavior in MOOCs and applied cluster analysis in order to investigate the connections between ATF usage and learning patterns.

1.3 ATF Effectiveness in MOOCs for Programming

MOOCs for programming have the potential to teach programming to a broad and diverse audience [26]. The high demand for computer professionals have led to an abundance of courses, with large numbers of enrollees [24]. Independent programming learning, however, is challenging. In addition to learning the programming principles and syntax of the language, code assignments pose a significant difficulty, especially in MOOCs where assistance from faculty or peers is scarce. Hence, automated feedback is of particular importance, with the potential of supporting learners, prevent frustration and even dropout [24]. Moreover, the flexibility of practicing and receiving feedback at any time is appropriate to the nature of the MOOC's learning [31]. The majority of studies on ATF focus on frontal courses, or online courses offered as part of a curriculum. It is likely that students in these courses interact extensively with the faculty, which enhances their learning [34], and might "overshadow" the impact of ATF on learning outcome [17]. In MOOCs, the impact of ATF system may be more significant. Yet, the effect of ATF on learning in MOOCs is under studied.

Currently, most research on automated feedback in MOOCs focuses on increasing error detection and feedback accuracy, with few reported on future intention to investigate the impact of the suggested ATF on learning [24, 27]. In other studies, factors to consider when developing ATF systems for MOOCs have been discussed, but no empirical results were presented [36]. According to a several studies, ATF is perceived by learners as improving performance and increasing engagement [7, 25]. The researchers [14] suggested that learners who formally registered to an ATF system were more engaged when solving code assignments than those who used the system partially, but not formally. No differences in performance or completion rates were observed. To summarize, there seems to be some evidence to indicate that automated feedback has the potential to support learners and enhance learning success in MOOCs for programming. Yet, there is still a lack of empirical research and a comprehensive picture of how the system is affecting learning behavior and outcomes.

1.4 Research Questions

In order to harness the potential of ATF in MOOCs, it is necessary to gain a better understanding of how the system influences learning behavior. Using a quantitative approach and an empirical design, the current study examines the relationship between ATF use and learning patterns in a MOOC, referring to relevant measures of learning in MOOCs. We suggest a comprehensive picture of learners' behavior, combining data of ATF usage, learners' interactions with course materials and their perception of the effect of ATF on learning. To that end, we pose the following research questions:

RQ1: Are the characteristics of learning behavior related to the interaction with course materials similar to those of ATF usage?

RQ2: What are the connections, if any, between the patterns of learning behavior and learners' responses to the automated feedback on code assignments?

RQ3: What is learners' perception with regard to the impact of ATF on learning?

2 Setting

2.1 The Course and ATF System

Our research field is a MOOC to learn the Python programming language, offered on Edx-based platform for MOOCs. The course was designed for beginners and no prior background in programming or Python is required. It consists of nine learning units, from the basics of programming in Python to the use of functions, data structures and working with files. The content is delivered through videos, in which short ungraded comprehension questions are embedded. Each unit includes closed exercises (e.g. multiple choice or text fill-in exercises, referred to as CE hereafter), answering of which is followed by an indication of correct/incorrect answer and a numeric grade. In addition, in order to provide learners with hands-on experience, code-writing assignments of different difficulty levels are offered. Programs ranging from a few lines of code to several dozen lines are required as solutions. To get the most out of the practice, learners are encouraged to submit their code solutions to the ATF system integrated into the course.

The system we implemented is INGIInious, an open-source software, supporting several programming languages and suitable for online courses (for more details on INGIInious, see [11, 19]). Upon submission, the INGIInious runs the code against a predetermined set of test scenarios and provides an instant feedback message, consisting of a grade and a textual component. Adapted to each assignment and error-type, the text may include varying levels of feedback (e.g. correct/incorrect, expected correct answer or more elaborated feedback), as classified by [35]. The system is incorporated into the course as an external tool, and registration is necessary for access. It is configured to allow unlimited re-submission of solutions.

Each cycle of the course is open for learning for six months. All course resources are available upon enrollment, enabling a self-paced mode of learning. It is offered free of charge, although a certificate can be earned for a small fee. Learners interested must, in addition to paying the fee, complete 70% of the closed exercises and submit a concluding project, with a weighted grade of 70 (out of 100). The course staff review the project and provides written feedback.

2.2 Population

The data for the present study were collected during the course cycle of June-December 21'. The research population consists of all learners who registered to the ATF system and submitted code-assignments at least once ($N = 899$). Among them, 655 (72.86%) filled out a demographic questionnaire. In terms of gender distribution, 73.28% of respondents identified as male, 26.57% as female and 0.15% as non-identified. The reported age ranged between less than 11 to over 75, with 15.57% under the age of 18, the majority (66.26%) in the range of 18–34 and 18.17% above. Based on self-reported prior knowledge, 32.67% of respondents had programming skills but did not know Python, 15.57% had prior Python knowledge, and 52.21% had no prior knowledge related to the course content.

3 Method

3.1 Operational Measures of Learning Behavior

In the context of the current study, learning behavior consists of engagement, persistence and performance (Table 1):

- **Engagement** is measured using variables related to watching videos, completing closed exercises and submitting code-assignments.
- **Persistence** is determined by the number of “touched” units, i.e. the number of units a learner interacted with video or a closed exercise or submitted a code-assignment.
- **Performance** is defined by the mean grade of closed exercises and the mean grade of code-assignments. The highest grade achieved in all attempts for each exercise or assignment was considered.

3.2 Data Resources and Pre-processing

It is one of the main goals of this study to present a comprehensive picture of learners’ behavior in the course. Therefore, we have gathered and analyzed data from multiple sources, as follows:

1. **Learning Activity Log**, including all events of learner’s interactions with course material. We pulled out three types of event: playing video, answering of comprehension questions, and attempts to answer closed exercises. Video replays for the same learner within the same video have been reduced to one event.
2. **ATF System Log**, containing records of code submissions. Each record includes the submitter ID, the submitted code, testing results and the generated feedback.
3. **Learners’ Responses to Self-reported Questionnaires**. Two questionnaires were administrated: one for demographic details including age, gender, and prior knowledge of programming and Python. The second one, titled as “learning experience”, collected learners’ perspectives of the impact of ATF on learning. Using a 5-point Likert scale, learners were asked questions about system’s contribution to engagement and learning effectiveness (e.g. “The system contributed to the motivation to complete more tasks in the course”).

The research was conducted under the rules of ethics, while protecting privacy and maintaining the security of information, and in accordance with the approval of the university ethics committee.

3.3 Definition of “Response to Feedback”

In order to obtain a learner’s response to feedback on a particular submission, we compared two consecutive submissions of the same code-assignment [32]. Three response types were defined: *any improvement* (AI), meaning an error detected in a particular submission has been fixed in the next one; *no improvement* (NI), when the same errors appeared in two consecutive submissions, and *getting worse* (GW), where the score of

Table 1. Learning behavior calculated measures

Learning behavior	Learning component	Variable	Description
Engagement	Course materials	Watched video	Percent of watched videos (out of the 29 videos in the course)
		Watched units	Number of units in which at least one video was watched (0–9)
		Active-watched ratio	Ratio between the number of videos in which the learner solved comprehension questions and the total number of videos watched (0–1)
		Solved closed exercises (CE)	Percent of CE a learner attempted, out of the 39 CE in the course (0–100)
		Solved units	Number of units in which at least one closed exercise was attempted (0–9)
		Mean attempts in CE	Mean attempts per closed exercise
		ATF usage	Submitted assignments
	Submitted units		Number of units in which at least one assignment was submitted (1–9)
	Mean attempts in assignments		Mean of attempts per code assignment
	Persistence	Course resources and ATF	Units touched

(continued)

Table 1. (continued)

Learning behavior	Learning component	Variable	Description
		Max unit touched	The most advanced “touched” unit (1–9)
Performance	Course resources	CE grade	The mean grade in CE (0–100)
	ATF usage	Submission score	The mean score in submitted assignments (0–100)

the following submission was lower. An empty value was assigned as the response to feedback for the last submission of each assignment or in case only one attempt was made for an assignment by the learner. The degree of improvement in response to feedback for each learner was determined as follows:

$$\text{Positive Response to Feedback (PRF)} = \frac{\sum(\text{AI responses})}{(\text{AI} + \text{NI} + \text{GW})} \quad (1)$$

The PRF ranges from 0 to 1, and its complement to 1 reflects non-improved responses.

4 Data Analysis and Findings

4.1 Learning Behavior - A Comprehensive Picture (RQ1)

For the purpose of analyzing the connections between learning behavior in the various learning components, the forementioned variables (Table 1) were extracted for each learner and descriptive data were generated, summarized in Table 2. Examining the correlation of the variables representing interactions with course materials and those representing ATF usage revealed the following results: the mean percentage of solved CE and submitted code-assignments, as well as the mean number of solved units and submitted units, were found to be strongly correlated ($r(897) = .76$ and $r(897) = .82$, respectively, $p < .001$). Similarly, a strong positive correlation was found between the percent of watched video and submitted assignments ($r(897) = .63$, $p < .001$), although lower than the correlation between watched video and solved CE ($r(897) = .81$, $p < .001$).

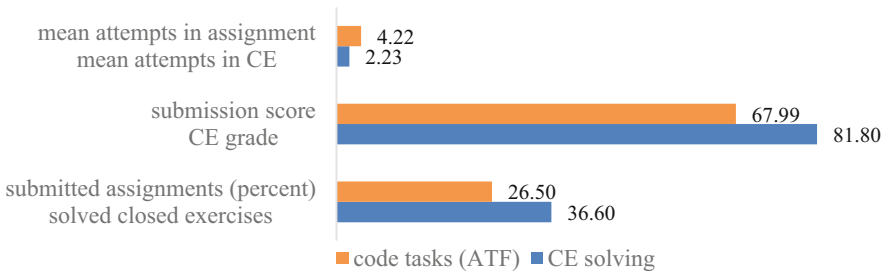
However, the mean grade on CE and the mean score on submissions were found to be weakly correlated ($r(897) = .22$, $p < .001$), while no correlation was found between the number of attempts in these two types of tasks. We further discuss this in Subsect. 5.

Even though the variables associated with solving CE and those associated with submitting code assignments correlated, the mean values of “paired” variables from these two sets differed significantly, as visualized in Fig. 1. A Shapiro-Wilk test of normality distribution was statistically significant, indicating a univariate normality deviation of learning behavior variables. Thus, the nonparametric Wilcoxon signed-rank test was used for the comparison. When compared to the percentage of code assignments learners

Table 2. Descriptive data of learning behavior variables (N = 899)

Learning component	Variable	Mean	SD	Mdn
Course materials	Watched video (percent)	41.40	30.70	34.00
	Watched units	4.17	2.88	3
	Active-watched ratio	0.38	0.23	0.4
	Solved CE (percent)	36.60	32.50	26.00
	Solved units	4.01	3.02	3
	Mean attempts in CE	2.23	1.45	2
ATF usage	Submitted assignments (percent)	26.50	32.20	8.00
	Submitted units	3.51	2.94	2
	Mean attempts in assignments	4.22	3.45	3.12
Course materials and ATF usage	Units touched	4.62	2.93	4
	Max unit touched	4.80	2.99	4
Course materials	CE grade	81.80	34.20	100
ATF usage	Submission score	67.99	35.43	82.76

submitted and the mean score they received for those assignments, more CE were completed, with higher grades achieved. The mean number of attempts per CE, however, was lower than the mean number of attempts per code assignment. Wilcoxon test indicated that these differences were statistically significant ($p < .001$).

**Fig. 1.** Learning behavior regarding solving CE and submitting code assignments

Cluster Analysis: Prior to clustering, PCA was applied to identify a subspace that carries the meaningful information with minimal redundancy (e.g. high-correlated variables) in the high-dimensional data in hand [5]. Five “differentiating” variables were identified, representing over 62.6% explained variance: watched video, submitted assignments, mean attempts in assignments, CE grade and submission score. K-mean cluster analysis was then performed with pre-defined number of five clusters, based on the elbow method plot and silhouette score [39]. The features of the clusters and mean values of differentiating variables are presented in Table 3.

Table 3. Identified clusters: mean values of five differentiating variables and max unit touched

	Name	Size	Watched video	Submitted assigns	Mean attempts per assign	CE grade	Submission score	Max unit touched
1	Mid-course learners	299	0.33	0.16	3.13	95.20	88.11	4.09
2	Completers, high performers	213	0.78	0.78	3.73	97.30	92.25	8.385
3	Content oriented mid-learners	189	0.36	0.06	3.20	94.00	21.34	4.063
4	Touched and left	123	0.15	0.05	4.01	20.00	50.95	2.472
5	Trail-error ATF users	75	0.28	0.11	12.87	87.00	64.44	3.173

The mean value of max unit touched was also calculated for each cluster, to add the persistence to the learning patterns observed. The clusters were named as follows: (1) “mid-course learners”: those who reached about the middle of the course, interacting to some extent with all course resources, and achieving fairly high grades. This is the largest group of learners. (2) “Completers, high performers”: learners with highest performance and completing rates, while medium submission rate per code assignment. This pattern was the second in number of learners. (3) “Content oriented mid-learners”: the third group in size, characterized by reaching to similar stage as the mid-course learners, while watching video content but rarely using the ATF system (may have solved code assignments without submitting to the system). (4) “Touched and left”: those who log in but showed almost no engagement with course materials and actually dropped out shortly after they started. (5) “Trail-error solvers”: those who submitted few code-assignments with many attempts, showing low persistence and performance. This was the least frequent behavior pattern.

4.2 The Response to Feedback (RQ2)

In examining the learners’ response to feedback, an interesting finding emerged, indicating that only in 36% of resubmissions, learners corrected the indicated error and resubmitted (mean PRF = 0.36, SD = 0.24, N = 796). Note that for learners who attempt only one solution per assignment (11.8% of learners), the PRF variable is empty as there was no consequent submission and thus no response to feedback. PRF was found to positively correlate with mean score on code assignments ($r(791) = .46, p < .001$), and negatively with mean attempts per assignment ($r(791) = -.25, p < .001$), suggesting that positive response to feedback shorten the way to correct solution.

Next, we compared PRF among the various clusters to examine how learners with different learning patterns responded to feedback. Levene's test indicated that the equality of variance assumption was not met, thus we use the non-parametric Kruskal Wallis test one-way ANOVA-by-Rank for the comparison [8].



Fig. 2. Mean values of PRF of the five learning behavior clusters (N = 791)

Findings suggest a connection between higher PRF and higher engagement and performance, where learners in the "Completers, high performers" cluster tend to correct and resubmit most often in compared to all other groups. The "mid-course learners" were next in line to fix errors and resubmit, whereas learners in clusters 3, 4, 5 were less likely to respond positively (Fig. 2). Kruskal Wallis test indicated statistically significant difference among the clusters regarding mean PRF ($H(4) = 196.64, p < .001$).

The differences were examined applying pairwise multiple comparisons using the nonparametric Dunn's test, which is suitable for unequal sample sizes such as cluster sizes in our case [40]. Significant difference was found between clusters 1 and 2 ($p_{\text{bonf}} = .003$), as well as between each of these two and each of the other three 3, 4, 5 ($p_{\text{bonf}} < .001$). No significant differences were found, however, among clusters 3, 4 and 5.

4.3 Learners' Perception of ATF Effects (RQ3)

We analyzed learners' responses to the "learning experience" questionnaire as supporting evidence, therefore applying descriptive statistics only. As indicated by 102 responses we received, learners tend to perceive that using the ATF system improves engagement, performance, and motivation for deeper learning. Treating "I strongly agree" and "I agree" (4 and 5 in Likert scale) as a consent, the majority of respondents agreed with the statements that the option to correct and resubmit prompted them to make an effort for a higher score (91.15%) and using the ATF system motivated them to be more engaged in solving CE and assignments (84.32%). Using the system enhanced coding skills, according to 84.31% of respondents, and 76.47% believed it enabled them to develop more correct solutions. According to 86.27% of those who responded, code testing and immediate feedback make learning more effective, and 84.31% found that the immediate feedback helped them progress more rapidly. Nevertheless, it is noteworthy that while the results indicate a positive impact of the system, about 53% of learners who answered the questionnaire completed eight or more learning units of the course, i.e. were characterized by high persistence and engagement.

5 Discussion

Regarding the first research question, positive correlations between variables associated with interactions with course materials and those related to ATF suggest that learners are generally consistent in their learning behavior. Those who consume content and solve closed exercises also choose to practice and submit code assignments. Yet, despite the similarity in trends, learners attempted and succeeded in solving more closed exercises relative to the number of code assignments submitted to the ATF and solved correctly. Referring to Bloom's taxonomy, [25] suggest that closed exercises assess only the degree of understanding of the main concepts while code assignments address higher and more complex levels of cognitive skills, thus being more challenging. The difference in learners' behavior regarding these two types of tasks may be explained, therefore, by their ability or determination to deal with the cognitive effort required for code assignments. Moreover, identifying and correcting errors in the code, as needed in code writing, is a difficult practice especially for beginners [10] and may result in increased number of resubmissions in comparison to solving close exercises.

Five clusters of learners with common learning behavior patterns emerged from the cluster. The identification of two groups of "extreme behaviors" - the "excelled" learners and those who dropout early, along with a third group of "mid-learners", is similar to results of previous studies applying clustering of MOOC learners (not specifically MOOCs for programming, e.g. [2]). Two additional groups were identified, based on their ATF usage patterns: those who reached half the course but rarely submitted code assignments ("content oriented mid-learners") and those exhibiting trial-and-error behavior in their ATF usage ("trial and error ATF users"). Combining these two data sources, i.e. course and ATF logs, enable us to characterize learners' behavior in more comprehensive way. To the best of our knowledge, this is the first study to use both course and ATF behavioral data for clustering.

Examine the effect of automated feedback on learning outcomes, as stated in RQ2, was one of the major goals of our study. Results offer evidence that a positive response to feedback (PRF) enhances the probability of reaching a correct answer, and even shortens the way until success. Less positive finding, however, is that in 64% of resubmissions the error pointed out by the ATF was not corrected, and the learner received the same feedback message again. An earlier study analyzing submissions for code assignments found a high percentage of non-improved submissions as well [28]. The loop of resubmitting and getting the same error-message can cause frustration and even dropout [37]. Adding the option to change the wording of feedback in a situation of identical repeated submissions may result in a "rescue" and a faster move towards a correct solution. In addition, identifying code assignments in which this phenomenon is particularly prevalent is recommended, to avoid potential attrition points in the course.

The connection between learning behavior and the response to feedback was demonstrated by comparing the value of PRF among the clusters we characterized. Findings indicated that learners in groups with lower level of engagement and persistence, and relatively low performance (clusters 3, 4, 5), responded positively less frequently, were unable to correct errors, or did not submit again. In contrast, however, the percentage of positive responses was highest among the "Completers, high performers" (cluster 2). Feedback has been found to be associated with higher performance in previous studies,

concerning frontal programming courses [16, 32]. Regarding the measures relevant to learning outcomes in MOOCs, our findings suggest that the positive response to feedback is significantly associated with success in the investigated MOOC.

As for RQ3, learners' perceptions regarding the impact of ATF on learning support the previous findings. In accordance with early studies both in the context of frontal and online programming courses (e.g. [30]) learners reported higher motivation for engagement in course assignments and considered the ATF as enhancing programming skills and learning effectiveness.

6 Conclusions and Future Work

In this study, we present a comprehensive picture of learning behavior in a MOOC for programming with an embedded ATF system. We believe that combining all the data into a single holistic picture is a significant contribution to advancing research in the field. Moreover, the indicated connections between ATF use and learning behavior may support the assumption that the automated feedback facilitates engagement, persistence, and performance. Nevertheless, we must be cautious in this context, and further research is needed to confirm the causal connection. It is primarily due to a limitation arises from the nature of the learning environment of the course, which includes an external interpreter enabling learners to actively solve code assignments, without receiving feedback, or having any indications in the analyzed data. Future research be undertaken with a setup allowing the comparison of these data as well, might bring additional insight into the effect of automated feedback. To maximize ATF effectiveness, however, exploring the causes of the high percentage of feedback-ignored resubmissions is suggested, as well as the impacts of feedback characteristics on learning behavior.

Acknowledgement. Our thanks to the Azrieli foundation for the award of a generous Azrieli Fellowship, which allowed this research. We thank the anonymous reviewers for their constructive comments.

References

1. Alario-Hoyos, C., Estévez-Ayres, I., Pérez-Sanagustín, M., Kloos, C.D., Fernández-Panadero, C.: Understanding learners' motivation and learning strategies in MOOCs. *Int. Rev. Res. Open Distrib. Learn.* **18**(3), 119–137 (2017). <https://doi.org/10.19173/IRRODL.V18I3.2996>
2. Anderson, A., Huttenlocher, D., Kleinberg, J., Leskovec, J.: Engaging with massive online courses. In: *WWW 2014 - Proceedings of the 23rd International Conference on World Wide Web*, pp. 687–697 (2014). <https://doi.org/10.1145/2566486.2568042>
3. Benotti, L., Aloï, F., Bulgarelli, F., Gomez, M.J.: The effect of a web-based coding tool with automatic feedback on students' performance and perceptions. In: *SIGCSE 2018 - Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, pp. 2–7 (2018). <https://doi.org/10.1145/3159450.3159579>
4. Cai, Y.-Z., Tsai, M.-H.: Improving programming education quality with automatic grading system. In: Rønningsbakk, L., Wu, T.-T., Sandnes, F.E., Huang, Y.-M. (eds.) *ICITL 2019*. LNCS, vol. 11937, pp. 207–215. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-35343-8_22

5. Carreira-Perpinán, M.: A review of dimension reduction techniques. Department of Computer Science. University of Sheffield. Technical report CS-96-09, pp. 1–69 (1997)
6. Cavalcanti, A.P., Barbosa, A., Carvalho, R., et al.: Automatic feedback in online learning environments: a systematic literature review. *Comput. Educ.: Artif. Intell.* **2**, 100027 (2021). <https://doi.org/10.1016/J.CAEAI.2021.100027>
7. Chan, M.M., De La Roca, M., Alario-Hoyos, C., Plata, R.B., Medina, J.A., Rizzardini, R.H.: MOOCMaker-2017 perceived usefulness and motivation students towards the use of a cloud-based tool to support the learning process in a Java MOOC. In: International Conference MOOC-MAKER, pp. 73–82 (2017)
8. Chan, Y., Walmsley, R.P.: Learning and understanding the Kruskal-Wallis one-way analysis-of-variance-by-ranks test for differences among three or more independent groups. *Phys. Ther.* **77**(12), 1755–1762 (1997). <https://doi.org/10.1093/ptj/77.12.1755>
9. Combéfis, S.: Automated code assessment for education: review, classification and perspectives on techniques and tools. *Software* **1**, 3–30 (2022). <https://doi.org/10.3390/software1010002>
10. Denny, P., Luxton-Reilly, A., Carpenter, D.: Enhancing syntax error messages appears ineffective. In: The 2014 Conference on Innovation & Technology in Computer Science Education, pp. 273–278 (2014). <https://doi.org/10.1145/2591708.2591748>
11. Derval, G., Gego, A., Reinbold, P., Benjamin, F., Van Roy, P.: Automatic grading of programming exercises in a MOOC using the INGIInious platform. In: European Stakeholder Summit on experiences and best practices in and around MOOCs (EMOOCs 2015), pp. 86–91 (2015)
12. Evans, B.J., Baker, R.B., Dee, T.S.: Persistence patterns in massive open online courses (MOOCs) **87**, 2, 206–242 (2016). <http://dx.doi.org/10.1080/00221546.2016.11777400>, <https://doi.org/10.1080/00221546.2016.11777400>
13. Feklistova, L., Luik, P., Lepp, M.: Clusters of programming exercises difficulties resolvers in a MOOC. In: Proceedings of the European Conference on e-Learning, ECEL, vol. 2020-October, pp. 563–569 (2020). <https://doi.org/10.34190/EEL.20.125>
14. Gallego-Romero, J.M., Alario-Hoyos, C., Estévez-Ayres, I., Delgado Kloos, C.: Analyzing learners’ engagement and behavior in MOOCs on programming with the Codeboard IDE. *Educ. Tech. Res. Dev.* **68**(5), 2505–2528 (2020). <https://doi.org/10.1007/s11423-020-09773-6>
15. Gordillo, A.: Effect of an instructor-centered tool for automatic assessment of programming assignments on students’ perceptions and performance. *Sustainability* **11**(20), 5568 (2019). <https://doi.org/10.3390/su11205568>
16. Gusukuma, L., Bart, A.C., Kafura, D., Ernst, J.: Misconception-driven feedback: results from an experimental study. In: ICER 2018 - Proceedings of the 2018 ACM Conference on International Computing Education Research, pp. 160–168 Association for Computing Machinery, Inc., New York (2018). <https://doi.org/10.1145/3230977.3231002>
17. Hao, Q., Wilson, J.P., Ottaway, C., Iriumi, N., Arakawa, K., Smith, D.H.: Investigating the essential of meaningful automated formative feedback for programming assignments. In: Proceedings of IEEE Symposium on Visual Languages and Human-Centric Computing, VL/HCC, pp. 151–155. IEEE Computer Society (2019). <https://doi.org/10.1109/VLHCC.2019.8818922>
18. Hew, K.F.: Promoting engagement in online courses: what strategies can we learn from three highly rated MOOCs. *Br. J. Edu. Technol.* **47**(2), 320–341 (2016). <https://doi.org/10.1111/bjet.12235>
19. INGIInious [software] (2014). <https://github.com/UCL-INGI/INGIInious>
20. Jung, Y., Lee, J.: Learning engagement and persistence in massive open online courses (MOOCs). *Comput. Educ.* **122**, 9–22 (2018). <https://doi.org/10.1016/j.compedu.2018.02.013>
21. Kahan, T., Soffer, T., Nachmias, R.: Types of participant behavior in a massive open online course. *IRRODL* **18**(6), 1–18 (2017). <https://doi.org/10.19173/irrodl.v18i6.3087>

22. Keuning, H., Jeurings, J., Heeren, B.: A systematic literature review of automated feedback generation for programming exercises. *ACM Trans. Comput. Educ.* **19**(1), 1–43 (2018). <https://doi.org/10.1145/3231711>
23. Kizilcec, R.F., Piech, C., Schneider, E.: Deconstructing disengagement: analyzing learner subpopulations in massive open online courses. In: *ACM International Conference Proceeding Series*, pp. 170–179 (2013). <https://doi.org/10.1145/2460296.2460330>
24. Krugel, J., Hubwieser, P., Goedicke, M., et al.: Automated measurement of competencies and generation of feedback in object-oriented programming courses. In: *2020 IEEE Global Engineering Education Conference (EDUCON)*, pp. 329–336. IEEE (2020)
25. Krusche, S., Seitz, A.: Increasing the interactivity in software engineering MOOCs—a case study. In: *Proceedings of the 52nd Hawaii International Conference on System Sciences*, pp. 7592–7601 (2019)
26. Luik, P., et al.: Participants and completers in programming MOOCs. *Educ. Inf. Technol.* **24**(6), 3689–3706 (2019). <https://doi.org/10.1007/s10639-019-09954-8>
27. Marin, V.J., Pereira, T., Sridharan, S., Rivero, C.R.: Automated personalized feedback in introductory Java programming MOOCs. In: *Proceedings - International Conference on Data Engineering*, pp. 1259–1270 (2017). <https://doi.org/10.1109/ICDE.2017.169>
28. McBroom, J., Yacef, K., Koprinska, I., Curran, J.R.: A data-driven method for helping teachers improve feedback in computer programming automated tutors. In: Penstein Rosé, C., et al. (eds.) *AIED 2018. LNCS (LNAI)*, vol. 10947, pp. 324–337. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93843-1_24
29. Narciss, S.: Feedback strategies for interactive learning tasks. In: Spector, J.M., Merrill, M.D., Van Merriënboer, J., Driscoll, M.P. (eds.) *Handbook of Research on Educational Communications and Technology*, pp. 125–144. Lawrence Erlbaum Associates, Mahaw, New York (2008)
30. Pettit, R., Prather, J.: Automated assessment tools: too many cooks, not enough collaboration. *J. Comput. Sci. Coll.* **32**(4), 113–121 (2017)
31. Pieterse, V.: Automated assessment of programming assignments. In: *3rd Computer Science Education Research Conference on Computer Science Education Research*, vol. 3, pp. 45–56 (2013). <http://dx.doi.org/10.1145/1559755.1559763>
32. Qian, Y., Lehman, J.: Using targeted feedback to address common student misconceptions in introductory programming: a data-driven approach. *SAGE Open* **9**, 4 (2019). <https://doi.org/10.1177/2158244019885136>
33. Rafique, W., Dou, W., Hussain, K., Ahmed, K.: Factors influencing programming expertise in a web-based e-learning paradigm. *Online Learn. J.* **24**(1), 162–181 (2020). <https://doi.org/10.24059/olj.v24i1.1956>
34. Restrepo-Calle, F., Ramírez Echeverry, J.J., González, F.A.: Continuous assessment in a computer programming course supported by a software tool. *Comput. Appl. Eng. Educ.* **27**(1), 80–89 (2019). <https://doi.org/10.1002/cae.22058>
35. Shute, V.J.: Focus on formative feedback. *Rev. Educ. Res.* **78**(1), 153–189 (2008). <https://doi.org/10.3102/0034654307313795>
36. Staubitz, T., Klement, H., Renz, J., Teusner, R., Meinel, C.: Towards practical programming exercises and automated assessment in massive open online courses. In: *Proceedings of 2015 IEEE International Conference on Teaching, Assessment and Learning for Engineering*, TALE 2015, pp. 23–30 IEEE (2015). <https://doi.org/10.1109/TALE.2015.7386010>
37. Vinker, E., Rubinstein, A.: Mining code submissions to elucidate disengagement in a computer science MOOC. In: *LAK22: 12th International Learning Analytics and Knowledge Conference (LAK22)*, pp. 142–151 (2022). <https://doi.org/10.1145/3506860.3506877>
38. Wong, J., Baars, M., Davis, D., Van Der Zee, T., Houben, G.J., Paas, F.: Supporting self-regulated learning in online learning environments and MOOCs: a systematic review. *Int. J.*

- Hum.-Comput. Interact. **35**(4–5), 356–373 (2019). <https://doi.org/10.1080/10447318.2018.1543084>
39. Yuan, C., Yang, H.: Research on K-value selection method of K-means clustering algorithm. *J* **2**(2), 226–235 (2019). <https://doi.org/10.3390/J2020016>
40. Zar, J.H.: *Biostatistical Analysis*. Prentice Hall, New York (1999)