Unfolding Learners' Response to Different Versions of Automated Feedback in a MOOC for Programming – A Sequence Analysis Approach

Hagit Gabbay, Anat Cohen
Tel Aviv University
hagitgabbay@mail.tau.ac.il anatco@tauex.tau.ac.il

ABSTRACT

In MOOCs for programming, Automated Testing and Feedback (ATF) systems are frequently integrated, providing learners with immediate feedback on code assignments. The analysis of the large amounts of trace data collected by these systems may provide insights into learners' patterns of utilizing the automated feedback, which is crucial for the design of effective tools and maximizing their potential to promote learning. However, data-driven research on the impact of ATF on learning is scarce, especially in the context of MOOCs. In the current study, we combine a theoretical framework of feedback with educational data mining methods to investigate the effect of feedback characteristics on learning behavior in a MOOC. Sequence pattern analysis is implemented to explore and visualize the actions taken by learners in response to feedback which composed of cognitive, meta-cognitive, and motivational elements. We applied our research approach in an empirical design which consists of five cohorts (total over 2200 learners) utilizing different versions of ATF. The findings suggest that learners tend to adopt learning strategies in response to feedback and exhibit a preference for utilizing example solutions, while still coping with the challenge of solving the assignments independently. The impact of feedback function, content and structure is discussed in light of a detailed view of the differences as well as common trends in learning paths. Allowing for fine-grained insights, we found our research approach contributes to a more comprehensive understanding of the effect of automated feedback characteristics in MOOCs for programming.

Keywords

Automated feedback, MOOCs for programming, educational data mining, sequence pattern analysis

1. INTRODUCTION

Automated Testing and Feedback (ATF) systems, integrated in programming courses, provide learners with immediate feedback on code assignments and allows for unlimited resubmissions. Research suggests that incorporating an ATF system into a programming course is perceived by learners as beneficial for their learning and motivation [2, 16]. Yet, research on the system's effect on overall course outcomes has yielded inconclusive results and studies identified the main impact of the system in task-level (i.e. throughout solving code assignments) [1, 5, 20]. This may be a result of the complex nature of the feedback's effect, which is multifaceted and contingent upon various factors, including the feedback design [32]. In Massive Open Online Courses (MOOCs) for programming, characterized by large numbers of learners and self-directed learning, there is potential for ATF systems to assist learners and compensate for the lack of available instructor support. To design the automated feedback in MOOCs to maximize its effectiveness, it is necessary to understand how learners utilize the feedback and how feedback features affect learning. However, there is a lack of empirical studies on the impact of automated feedback characteristics on learning [20, 50], particularly in the context of MOOCs.

The asynchronous and self-paced nature of MOOCs poses challenges for instructors seeking to monitor and evaluate learners' utilization of the ATF system. In particular, it is difficult to determine the effectiveness of the feedback elements. Analyzing the large amounts of trace data collected by ATF systems may provide insights into learners' patterns of utilizing the feedback. However, data-driven research on the impact of automated feedback features on learning in MOOCs is scarce.

Addressing these gaps, the aim of the current study is to explore the effects of automated feedback characteristics on learning behavior in a MOOC for programming. To do so, we compare the behavior patterns of learners utilizing different versions of an ATF system, composed of cognitive, meta-cognitive, and motivational elements, within a MOOC. A data-driven approach is employed, consists of sequence pattern mining and statistical analysis. Notable, the assessment of our research approach is another goal of this study.

Sequence pattern mining (SPM) is a prevalent method within the domain of educational data mining for uncovering patterns in the sequential interactions of learners with educational systems [3]. By identifying patterns in the order and timing of learners' actions, SPM can provide valuable insights into learning strategies and behavior, which can be used to adapt and improve educational environments [4]. Unlike other methods, such as process mining, sequence mining is particularly well-suited for high-resolution analysis of learning behavior "at the local level", such as solving assignments [8]. Previous studies applied SPM to analyze learners' interaction with different course materials, examine learning behaviors during different periods of learning or identify different sequence patterns between predefined groups in different research conditions (e.g. [12, 39]). The process of solving a code assignment involves sequential actions taken by the learner over the time period allocated for the task. Characterized by order and timing, these sequences of actions reflect the pattern of interaction between the learner and the ATF system. Given our goal to compare behavior patterns in response to different versions of the ATF, the SPM method may be a suitable and applicable approach.

H. Gabbay and A. Cohen. Unfolding learners' response to different versions of automated feedback in a mooc for programming – a sequence analysis approach. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 160–170, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115711

2. RELATED WORK

2.1 Research framework for feedback effectiveness

The framework proposed by Narciss suggests that feedback can be characterized by three key factors, namely its function, content and presentation, all of which impact its effectiveness [32]. The function of feedback corresponds to the facet(s) of competencies it seeks to enhance, and can be classified as cognitive, meta-cognitive and motivational [33]. Cognitive feedback is aimed at promoting high-quality learning outcomes and the acquisition of the knowledge and cognitive operations necessary for accomplishing learning tasks (e.g. [31]). Metacognitive feedback directs the student's awareness of and ability to choose appropriate learning strategies [32], while motivational feedback may encourage students in maintaining their effort and persistence [33].

Feedback content is the information provided to the learner, which addresses the selected function. The content varies in terms of level of detail, as classified by [32]: basic feedback which only provides knowledge of result (KR), and sometimes knowledge of the correct response (KCR), or elaborate feedback (EF) providing additional information. [21] specified subtypes of elaborated feedback to classify the content of feedback on programming assignments, which can be identified in ATF systems. Among these, the relevant types for the current study include knowledge about mistakes (KM), such as test-failure errors and compiler errors; knowledge about metacognition (KMC), which relates to metacognitive feedback; and knowledge about how to proceed (KH), such as hints or examples. In the current study, we refer to the included components (e.g. text, hints or examples) as feedback structure. The content and structure both convey feedback function(s).

The presentation of feedback pertains to the way in which the content is presented or communicated to the learner, e.g. the timing, number of attempts, adaptability, or modality [32]. In ATF systems the feedback is commonly immediate, while the allowed number of attempts and the level of adaptivity, as well as the visuality of the provided feedback, may vary according to the system's characteristics and pedagogical approach [21, 35].

2.2 The impact of Feedback features

In the study in hand the focus is on the impact of the function and content of feedback, with no consideration of the presentation factor. Therefore, we consider feedback’s features as the function, content, and structure.

In a comprehensive literature review, [21] revealed that most systems provide cognitive feedback, typically in the form of information about errors and, occasionally, guidance on how to progress. In studies comparing the effectiveness of different types of feedback, [19, 20] suggest that correct/incorrect feedback (KR) is relatively less effective compared to KCR and EF. Their findings revealed that students who were provided with higher levels of feedback (KCR, EF) outperformed those who received only KR feedback across three complex programming assignments. Furthermore, the provision of KR feedback caused learners to make more attempts per assignment.

The impact of providing hints and/or example solutions, in various forms, has been examined in several studies. [19, 50] have found that providing students with fixed content hints as a help option has no significant impact on problem-solving performance. On the other hand, [28, 29] suggested that adaptive next-step on-demand hints have a positive effect on students' performance and learning. Yet, providing adaptive hints could be either computationally expensive or require great attention from human instructors and it should be considered whether it is cost-effective [20]. Example solutions, provided to learners in the form of hints during various stages of the problem-solving process, were indicated as more effective than other forms of feedback by [1, 34, 50]. The question of how learners react when they are given a choice between several help options has not yet been sufficiently investigated [18].

Metacognitive feedback was found only in few ATF tools [21], with inconclusive effects. [46] have shown positive results in the effect of metacognitive feedback on the strategy of collaborative learning. In contrast, [13] investigated the effect of explainable feedback on changes in learning strategies but no significant effect was found. A study in different knowledge domain found that immediate metacognitive feedback can help students acquire better help-seeking skills within an intelligent tutoring system for geometry. Moreover, the improved help-seeking skills transferred to learning new domain-level content [41]. Despite the limited research on the effect of metacognitive feedback in ATF, developing learning strategies is crucial for MOOC learners [44] thus we find it worth exploring this approach further.

Motivational feedback is also less common and in certain systems it combined with other forms of feedback [7]. Studies have demonstrated an effective use of motivational feedback, provide students with immediate positive feedback on completed objectives [45] or supportive motivational messages triggered by log data analysis [37]. [31] suggest a positive impact of motivational automated feedback on student engagement and performance in another knowledge domain.

In many cases, the feedback provided by ATF systems is not unidimensional and consists of more than one function, as well as a combination of several types of information [22]. Therefore, to investigate the impact of specific features, a comparative study is necessary, in which multiple versions distinctly differentiated in their features, are implemented simultaneously under consistent conditions pertaining to both the learning environment and the learners. With this approach and based on the proposed framework, we examined the impact of automated feedback features on learners' behavior patterns throughout solving code assignments. Of particular interest were the effects of detailed knowledge about mistakes (KM), hints, and example solutions (KH), as well as the provision of metacognitive and motivational functions which were less investigated.

2.3 Sequential pattern analysis of learning behavior

A variety of studies have been conducted to analyze the educational behaviors of students using sequence mining methods [43]. The objectives and applications of these studies vary widely, as well as the analytic approaches. For example, [24] conducted an analysis of MOOC log data of learners who completed final assessments, with the aim of comparing the behavioral patterns of learners with varying levels of achievement. The study employed SPM to extract frequencies of predefined sequences, representing engagement and time management behaviors. Subsequently, statistical analysis was performed to uncover distinctions among the groups. [23] utilized sequence mining techniques to identify differentially frequent patterns between distinct groups of students, without predefining patterns of learning. By utilizing performance data, segments of productive and unproductive learning behaviors were identified and compared. [4] analyzed data of MOOC for learning programming principles, to investigate study patterns exhibited by learners during assessment periods and the evolution of these patterns over time. Sequence mining methods were applied in two approaches, with predefined patterns and in an unsupervised manner, to capture study patterns from learners' interaction sequences.

For more comprehensive examination, recent studies use sequence pattern analysis in conjunction with other EDM methods (e.g. process mining) to identify and investigate learning tactics and strategies [3, 11, 25, 40] or to construct prediction models for learners' behavior such as dropout or course completion [8]. In order to gain insights of the way learners utilize ATF system, [30] analyzed records of submitted code assignments and clustered programs with similar functionality. SPM was then applied to trace student progress throughout an exercise. However, the effect of feedback characteristics on learners' behavior was not explored.

3. RESEARCH QUESTIONS

The present study aims to address the research gaps concerning the effect of automated feedback features in the context of MOOC for programming. With a data-driven approach, we employ sequence pattern analysis to explore and compare learners' response to various forms of feedback, which differ in terms of feedback features: function, content and structure. In this regard, the following research questions were posited:

RQ1: How do feedback features affect learners' behavior patterns throughout solving code assignments in a MOOC for programming? In particular,

RQ2: What is the impact of feedback features on the usage of the various help forms?

RQ3: What are the characteristics of utilizing hints and example solutions, and to what extent do they contribute to advancing the correct solution?

In an implementation perspective, our research approach may provide instructors with fine-grained insights of the utilization of ATF tools in MOOCs for programming, in order to maximize their contribution to learning. Therefore, the assessment of this approach is another goal of this study.

4. METHODOLOGY

To explore the connections between feedback features and learners' response we integrated an ATF system into a MOOC for programming and developed five different versions of the feedback provided by the system.

4.1 The ATF system

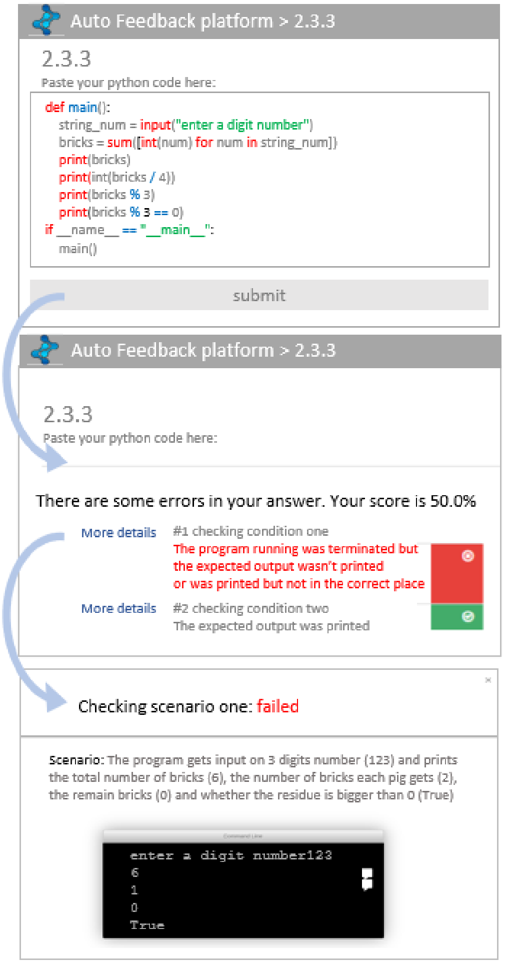
The integrated system is INGInious – an open-source software, supporting several programming languages and suitable for online courses (for more details see [10]). Applying static checks and running pre-defined tests for each assignment, the system provides immediate feedback, consists of a grade and a textual message. Error types detected are syntax errors, incorrect implementation of instructions, exception errors (prevent the code from being executed) and test-failed errors, where the results do not match the expected ones. For each type of error in each assignment, the feedback can be customized in advance to include an appropriate text and additional objects such as hints or example solution. In the current study, we used this configurability to explore learners' behavior in response to different feedback versions.

4.2 Feedback Versions

Based on the framework suggested by Narciss [32], five versions of the ATF system were designed, each with a different feedback function, content and structure. We tailored the feedback for the various error types to reflect the differences between the versions:

(1) **The Base version (V-Base)** provides the compiler message as-is for syntax errors. For exception and test-failed errors, the feedback includes a description of what should have been executed or output. An optional help form, which appears as "More details" (referred to as HMD from here on), is available. It consists of the exact breakdown of the actual vs. expected outcome (Figure 1). Although this version can be classified as cognitive feedback [32], we consider it "cognitive-light", compared to the other versions.

(2) **The Enriched Cognitive version (V-EC)** provides more elaborate text for each error type, offering cognitive knowledge. In addition to the HMD, a hint is also available upon request (i.e. clicking on a link). The hints are predefined and formulated based on common errors for each assignment (similar to [19]). They guide towards the correct solution but are not adaptive. In case of in case of multiple requests, the same text is presented for the same assignment each time.


Auto Feedback platform > 2.3.3
2.3.3
Paste your python code here:

```
def main():
    string_num = input("enter a digit number")
    bricks = sum([int(num) for num in string_num])
    print(bricks)
    print(int(bricks / 4))
    print(bricks % 3)
    print(bricks % 3 == 0)
if __name__ == "__main__":
    main()
```

submit
Auto Feedback platform > 2.3.3
2.3.3
Paste your python code here:

There are some errors in your answer. Your score is 50.0%

More details #1 checking condition one
The program running was terminated but the expected output wasn't printed or was printed but not in the correct place

More details #2 checking condition two
The expected output was printed

Checking scenario one: failed

Scenario: The program gets input on 3 digits number (123) and prints the total number of bricks (6), the number of bricks each pig gets (2), the remain bricks (0) and whether the residue is bigger than 0 (True)

```
enter a digit number123
6
1
0
True
```

Figure 1. An example of the interface of the ATF system (V-Base version): the submitted code (top), the corresponding feedback message (middle), and the "More details" (HMD) option (bottom).

(3) **Meta-cognitive version (V-MC)** – knowledge of learning strategies, as defined by [41], is added to the text messages of V-EC. To enhance help-seeking strategies, learners are

encouraged to use the provided help forms and, in some cases, review the relevant content in the learning units. Additionally, after submitting a correct solution, an example solution is made available as a further learning strategy.

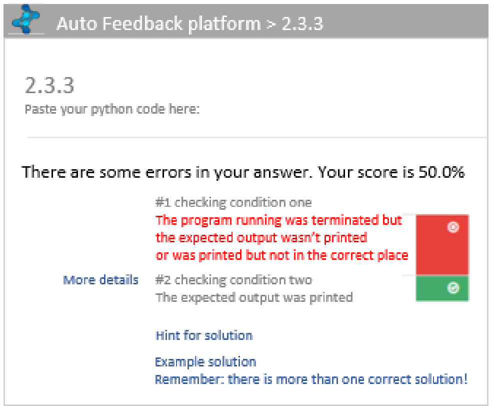
(4) **Motivational version (V-Motiv)** – feedback messages in V-EC are enhanced with positive and motivational language. Similar to [31], the text includes encouraging statements for partially correct solutions and overall motivating phrases.

(5) **Example solution version (V-ES)** - provides an option to view an example solution immediately following the initial attempt on an assignment, in addition to HMD and hints (Figure 2). Unlike other studies that recommend solutions for the next step or a similar assignment (e.g. [36, 47]), V-ES offers a complete solution for the current assignment.

Table 1 summarizes the functions and structures of the various versions and provides an example of the text presented.

Table 1. Function and structure of the five feedback versions

Version and function	Optional help forms	Examples of feedback text message (for test-failed errors) and hints
V-Base Cognitive "light"	HMD	*The program run was terminated but the expected output (…) wasn't printed or was printed but not in the correct place.*
V-EC Cognitive	HMD, hints	In addition to the text of V-Base: *The tested case is..* *The reason for the error may be.. or..* *Did you check…?* *Are the prints ordered correctly?*
		An example of a hint: *The input includes two types. Use if, else or elif to separate the input into two types (e.g. float/int or upper/lower case) and then run the appropriate conversion function.*
V-MC Cognitive + Meta-Cognitive	HMD, hints, example solution (Only after a correct solution)	In addition to the text of V-EC: *Use "More details" to check the received output.* Hints are the same as for V-EC.
V-Motiv Cognitive + Motivational	HMD, hints	In addition to the text of V-EC, if some case tests run correctly: *The program has shown to be successful in some test cases, indicating that you are headed in the right direction.* *Great job, you're making progress! Keep working on making the solution compatible with all input types and resubmit.* Hints are the same as for V-EC
V-ES Cognitive	HMD, hints, example solution	Text messages and hints are the same as for V-EC



Auto Feedback platform > 2.3.3

2.3.3

Paste your python code here:

There are some errors in your answer. Your score is 50.0%

More details

#1 checking condition one

The program running was terminated but the expected output wasn't printed or was printed but not in the correct place

#2 checking condition two

The expected output was printed

Hint for solution

Example solution

Remember: there is more than one correct solution!

Figure 2. Enriched cognitive feedback with HMD, hint and example solution options, V-ES version.

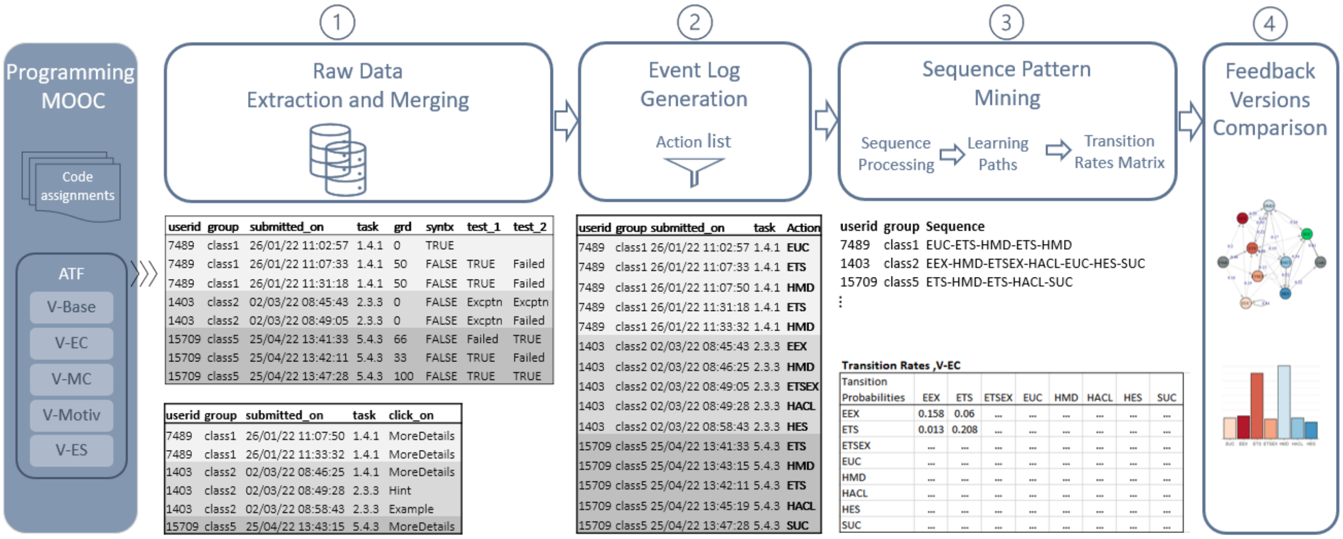
4.3 Research Field and data set

Our research field is a MOOC for learning the Python programming language, which is offered on an Edx-based platform. The course covers a range of topics organized into nine units, from the very basics of Python to the use of data structures, file manipulation, and functions. The course contains 29 video lectures, 39 exercises, and 53 code assignments, which are of varying levels of difficulty. It follows a self-paced learning mode, with all course materials available at once and no set deadlines. 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.

The INGInious ATF system with the five feedback versions was incorporated into the course schedule between January to July 2022. It was incorporated as an external tool and configured to allow for unlimited submissions, aligning with MOOC learning concepts [35]. The cohort-mechanism embedded in the Edx platform was utilized to randomly assign learners enrolled in the course to one of five groups, each of which had access to a different feedback version. The allocation was carried out during registration and was fixed for the duration of the course, thus precluding any transfer between groups and ensuring that each learner was exposed exclusively to a single version of feedback.

The usage of the ATF system was voluntary. Of 16,602 enrollees, 2206 learners (13.3%) chose to register and use the system. Demographics provided by 75% of these learners, with 28% female, 71.5% male, and 0.5% "other". The age range varied from less than 11 years old to over 75 years old, with 11.17% under the age of 18, the majority (69.41%) between the ages of 18 to 34, and 19.42% above 34 years old. 58.7% had no prior knowledge in programming, 29.3% had knowledge in other programming languages, and 12% had prior knowledge in Python. A chi-squared test confirmed equal variance of gender, age and prior knowledge among the five experimental groups of the ATF users.

Data resources consist of ATF log files, including 165,282 submission records and 57,556 records of clicks on help forms offered in the various feedback versions ("help-clicks"). In this study, we analyzed a subset of the data, which only includes the actions of the learners when solving the four assignments in Unit 4 (11,519 submission and 8,769 help-clicks records). 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.



userid	group	submitted_on	task	grd	syntx	test_1	test_2
7489	class1	26/01/22 11:02:57	1.4.1	0	TRUE		
7489	class1	26/01/22 11:07:33	1.4.1	50	FALSE	TRUE	Failed
7489	class1	26/01/22 11:31:18	1.4.1	50	FALSE	TRUE	Failed
1403	class2	02/03/22 08:45:43	2.3.3	0	FALSE	Excptn	Excptn
1403	class2	02/03/22 08:49:05	2.3.3	0	FALSE	Excptn	Failed
15709	class5	25/04/22 13:41:33	5.4.3	66	FALSE	Failed	TRUE
15709	class5	25/04/22 13:42:11	5.4.3	33	FALSE	TRUE	Failed
15709	class5	25/04/22 13:47:28	5.4.3	100	FALSE	TRUE	TRUE

userid	group	submitted_on	task	click_on
7489	class1	26/01/22 11:07:50	1.4.1	MoreDetails
7489	class1	26/01/22 11:33:32	1.4.1	MoreDetails
1403	class2	02/03/22 08:46:25	1.4.1	MoreDetails
1403	class2	02/03/22 08:49:28	2.3.3	Hint
1403	class2	02/03/22 08:58:43	2.3.3	Example
15709	class5	25/04/22 13:43:15	5.4.3	MoreDetails

userid	group	submitted_on	task	Action
7489	class1	26/01/22 11:02:57	1.4.1	EUC
7489	class1	26/01/22 11:07:33	1.4.1	ETS
7489	class1	26/01/22 11:07:50	1.4.1	HMD
7489	class1	26/01/22 11:31:18	1.4.1	ETS
7489	class1	26/01/22 11:33:32	1.4.1	HMD
1403	class2	02/03/22 08:45:43	2.3.3	EEX
1403	class2	02/03/22 08:46:25	2.3.3	HMD
1403	class2	02/03/22 08:49:05	2.3.3	ETSEX
1403	class2	02/03/22 08:49:28	2.3.3	HACL
1403	class2	02/03/22 08:58:43	2.3.3	HES
15709	class5	25/04/22 13:41:33	5.4.3	ETS
15709	class5	25/04/22 13:43:15	5.4.3	HMD
15709	class5	25/04/22 13:42:11	5.4.3	ETS
15709	class5	25/04/22 13:45:19	5.4.3	HACL
15709	class5	25/04/22 13:47:28	5.4.3	SUC

userid	group	Sequence
7489	class1	EUC-ETS-HMD-ETS-HMD
1403	class2	EEX-HMD-ETSEX-HACL-EUC-HES-SUC
15709	class5	ETS-HMD-ETS-HACL-SUC
⋮		

Transition Rates ,V-EC

Tansition Probabilities	EEX	ETS	ETSEX	EUC	HMD	HACL	HES	SUC
EEX	0.158	0.06	...	...	...	...	...	...
ETS	0.013	0.208	...	...	...	...	...	...
ETSEX	...	...	...	...	...	...	...	...
EUC	...	...	...	...	...	...	...	...
HMD	...	...	...	...	...	...	...	...
HACL	...	...	...	...	...	...	...	...
HES	...	...	...	...	...	...	...	...
SUC	...	...	...	...	...	...	...	...

Figure 3. Analytic approach to uncover learning patterns and compare the response to feedback among experimental groups.

4.4 Analytic approach to detect patterns of learning behavior

To explore learning behavior in fine-grained manner we utilized SPM, analyzing patterns of actions taken by learners throughout solving code assignments. Based on variables gathered from the sequence analysis, statistical tests were conducted to compare between the five experimental groups.

The method we applied to identify sequence patterns is similar to those described in previous studies [11, 43] (Figure 3). First, raw data of submission and help-click records for each learner were extracted and merged, based on assignment id and time stamps (1). Using a list of action codes, composed to represent learner actions (detailed below), we generated the action-event log (2). The exploratory sequence analysis implemented in the TraMineR package of R [15] was then utilized (3) to produce sequences of actions for each assignment and learner, as well as compute transition probability matrices and uncover characteristics of learning paths. Finally, we visualized and compared behavior patterns, using the five experimental groups as designated clusters (4).

4.4.1 Response Actions and definitions

Upon receiving feedback, a learner may resubmit a revised solution, use help, or waive the assignment without making any further submissions. To identify the differences between consecutive submissions, the "resubmitting" action is represented by several specific actions, based on error-types detected by the ATF system (section 4.1). The response-action list used for sequence analysis is described in Table 2.

4.4.2 Sequence processing

The action-event log, produced by converting the raw submission data, was transformed into sequences of actions for each assignment and each learner attempted. Sequences of length 1 (one submission to the assignment) were removed, having no evidence of feedback effect. The same applied to sequences including only repeated SUC actions.

Repeated identical actions of opening a hint or an example solution (HACL or HES) have been replaced with a single action, since the hints and example solution are fixed (for each assignment). To obtain a more representative dataset, very long sequences (higher than the 95th quantile) were removed. The response time for resubmission, defined as the period of time between consecutive submissions, was calculated for each pair of submissions. Response time longer than IDLE TIME (determined as 10 minutes) were replaced with IDLE TIME value.

Table 2. Response-action codes and definitions

Action type	Action code	Description
Submitting Actions	SUC	A correct solution.
	EUC	Code with syntax error.
	EIN	Code with instructions mismatch
	EEX	Code with exception for all test cases (can't be executed).
	ETS	Code that fails in some/all cases (the results differ from the expected ones)
	ETS-EX	Code with a mix of exceptions and test-failed errors
Help usage actions (help-clicks)	HMD	Clicking on "More details".
	HACL	Clicking on "hint"
	HES	Clicking to open the example solution.
"Fake" actions	Fsub, Lsub	First and last actions in sequence, added for visualization

4.4.3 Sequence analysis

A typical sequence consists of code-submission actions and help-usage actions (Figure 4). To explore the behavior patterns of learners, we analyzed the produced sequences from two perspectives: (1) The frequency of a specific action or a group of actions within the entire actions performed by a group of learners (e.g. the frequency of help-clicks in sequences 6-7 is 0.42) and (2) The percentage of learners who performed a specific action or subsequence of actions out of a group of learners (e.g. 29% of learners performed the EEX action (two out of seven)). This measure is the support of a pattern of actions.

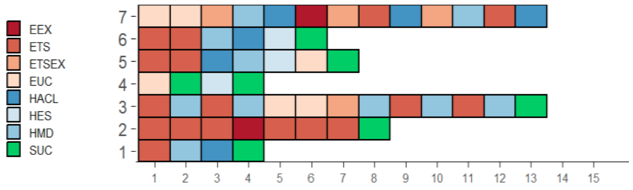


Figure 4. Example of sequences produced, representing the actions taken by learners when solving an assignment (each sequence represents the actions of a single learner).

5. RESULTS

In order to assess the effect of the feedback characteristics and evaluate the usability of our approach, we applied it to analyze the data of learners' interactions with the ATF system while completing four assignments in unit 4 of the course. These assignments are relatively homogeneous in terms of difficulty level and distribution of common error types, with a sufficient number of learners in each group who submitted more than once (a total of 567 learners and over 20,000 submission and help-click records). Data was analyzed for each assignment separately and for all four assignments together. We used non-parametric Kruskal-Wallis and Dunn tests to compare the relevant variables between groups, as the assumptions for one-way ANOVA were not met [6]. The reported p-values adjusted with the Bonferroni method.

5.1 RQ1: The Effect of Feedback Features on Learning Patterns

The analysis of learners' sequence of actions revealed significant differences in patterns between some of the experimental groups, as well as similarities among others. The average number of actions performed by learners (in all four assignments), represented by mean sequence length, was higher for groups V-MC and V-EC and lower for V-ES and V-BASE (Table 3).

Table 3. Sequence characteristics: length, time between actions and total duration (minutes). Higher values in red, lower values in blue.

Version (learners in group)	Sequence Length Mean (SD)	Time between actions, Mean (SD)	Time on Assign., Mean (SD)
V-Base (95)	10 (6.65) [1]	1.33 (0.91)	14.18 (13.5)
V-EC (112)	12.6 (7.92) [2]	1.40 (0.74) [3]	18.37 (14.74)
V-MC (116)	12.8 (8.45) [2]	1.31 (0.75)	17.45 (14.75) [4]
V-Motiv (121)	11.2 (8.81) [2]	1.44 (0.82) [3]	17.08 (14.84) [4]
V-ES (123)	9.51 (6.65) [1]	1.23 (0.79)	12.4 (11.30)
Kruskal-Wallis	$x^2(4) = 299.03$ p<.001	$x^2(4) = 141.34$ p<.001	$x^2(4) = 337.62$ p<.001

[1] no significant diff. between V-Base and V-ES
[2] no significant diff. between V-EC, V-MC and V-Motiv
[3] no significant diff. between V-EC and V-Motiv
[4] no significant diff. between V-EC and V-Motiv

The time intervals between actions, which may indicate the time spent processing feedback and revising the solution, were found to be longer, on average, for learners in V-EC and V-Motiv groups and shorter for learners in V-ES group. Additionally, the mean duration from first to last action in the sequence, representing time spent on the assignment, was highest among V-EC learners and lowest among V-ES learners. The distribution of the actions related to submitting assignments was consistent among all groups, as indicated by similar frequencies of EUC, EEX, ETS and ETS-EX detected in submissions. The percentage of learners who reached the correct solution, measured by the frequency of SUC action, was similar as well, and close to 100%. Thus, no difference was found regarding the score on assignments.

5.2 RQ2: The Impact of Feedback Features on the Usage of the Help Forms

Differences were observed between the groups in terms of help-usage actions. The variety of help forms offered to each group affected the overall use of help, as shown in Figure 5. With 0.34 of help-usage actions out of all actions, learners in the V-ES group used help more frequently compared to learners in the other groups ($x^2(4) = 24.78$, p<.001). V-Base showed the lowest usage of help, with 0.25 (only HMD in this case). Learners in V-EC and V-Motiv groups showed similar behavior in this manner, with 0.28 of help-use actions.

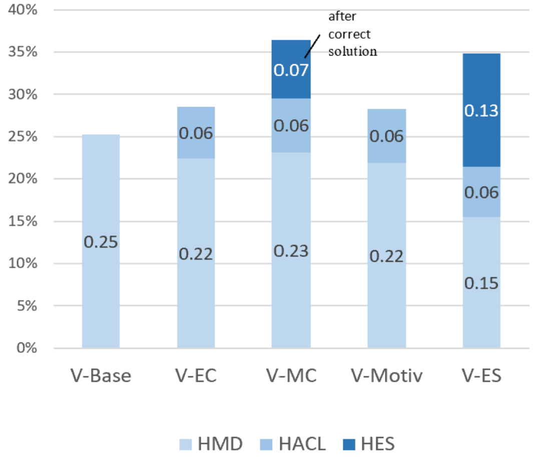


Figure 5. Frequency of help-usage actions performed by learners in each group

V-MC group utilized HES in 0.07 of actions but only after submitting a correct solution. Thus, their pattern of help-usage while attempting seems to be at medium level as well (0.29). Nevertheless, if we exclude these HES actions, the proportion of HMD actions during the solving process (before a correct solution is submitted) becomes 0.25, and the overall help usage of the V-MC group rises to 0.31. Although these adjusted values have not been found to be statistically significant, they suggest that learners in V-MC group tended to utilize the available help options (before submitting a correct solution) more frequently compared to V-EC and V-Motiv learners.

Upon further analysis, additional differences were found regarding the use of HMD, which is available in all versions in response to incorrect submissions. The likelihood of learners for this pattern of help seeking following EEX action reflects in the percentage of sequences containing the subsequence EEX-HMD at least once (Figure 6). The highest percent was found for V-MC group with 76% and the lowest for V-ES, with 43% (chi-squared independence, $x^2(4) = 14.664$, p<.01). The Pearson standardized residuals obtained (for using the HMD) are 2.39 for V-MC and -3.38 for V-ES. That is, learners provided with the metacognitive version responded to the explicit suggestion to use HMD after an exception error, as a learning strategy.

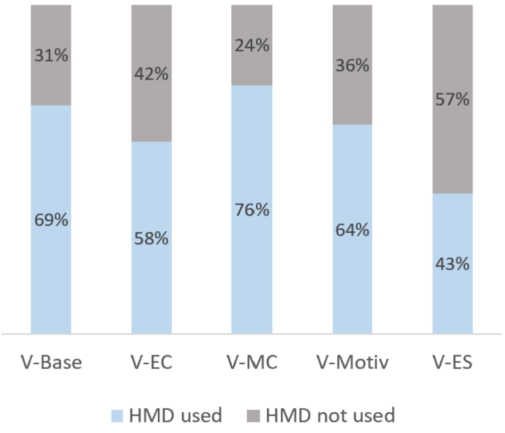


Figure 6. HMD usage after EEX (HMD-EEX proportional frequency)

Notably, the V-Base group showed close value of 69%, which is expected as HMD is the only help form provided in this version. A similar finding was obtained for the ETS-HMD sequence. The learners in the V-ES group were less likely to use HMD compared to the other groups while V-EC and V-Motiv groups exhibited comparable behavior.

5.3 RQ3: Patterns of Utilizing the Example Solutions and Hints

Calculating the probability matrix for transitions between states of each group allowed for detailed tracking of the use of help forms.

The example solution, as an aid to reach a correct solution, was offered only to learners in the V-ES group and was available right after the first submission attempt for an assignment. Approximately 75% of learners in this group utilized this form of help. Analysis of their action sequences revealed that the probability of transitioning **to** HES from one of the error states (EEX, ETS, ESTEX or EUC) was only about 0.34. This suggests that learners often first attempted to utilize other available forms of help, such as HMD or HACL, before turning to the example solution for assistance (Figure 7). **After** utilizing the example, the probability of resubmitting an incorrect code, signifying a transition from HES to one of the error states, was 0.31 (Figure 8). This outcome implies that in about one third of cases learners did not copy-paste the solution but tried to solve themselves. The likelihood of learners resubmitting a correct solution (i.e., transitioning from HES to SUC) was calculated to be 0.34, whereas the probability of seeking additional help (transitioning to HMD or HACL) was found to be 0.13. Only 0.22 of the cases resulted in learners choosing to waive and not resubmit. Taken together, these findings suggest that the availability of the example solution did not discourage learners in V-ES from trying to solve the assignment independently.

For V-EC learners, the example solution was offered as a mean of further learning, only becoming available after they submitted a correct solution. Therefore, it should not be considered as an aid to facilitate assignment completion. Approximately 58% of the learners in V-EC group chose to open the provided example.

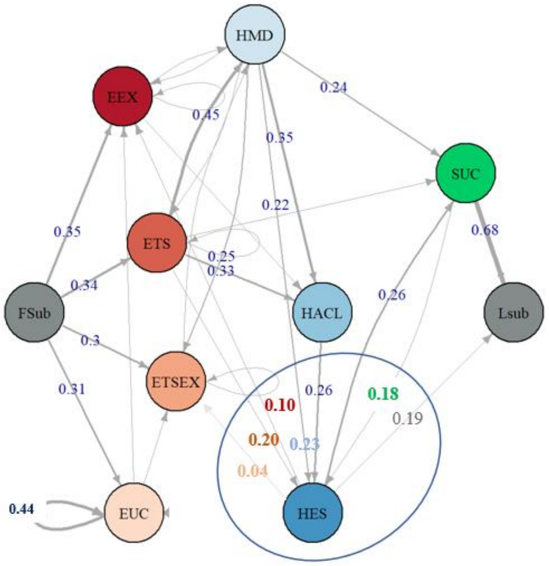


Figure 7. Transition probabilities for V-ES group. On edges: probability of transition <u>to</u> an action state (in particular, before opening HES). Probabilities less than 0.05 were omitted. Rates for HES are encircled and colored with the color of the "origin" state.

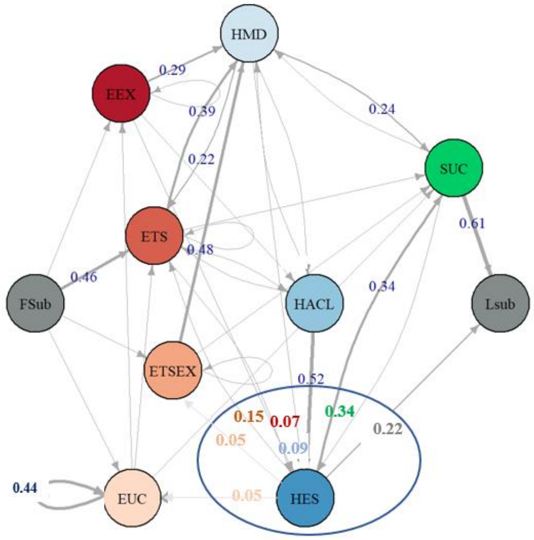


Figure 8. Transition probabilities for V-ES group. On edges: probability of transition <u>from</u> an action state (i.e. after opening HES). Probabilities less than 0.05 were omitted. Rates for HES are encircled and colored with the color of the "target" state.

The hint was provided to all groups of learners except V-Base. A quit low probability of 0.08 – 0.11 (with average of 0.09) was found for transitioning from EEX, ETS or ETSEX error-states to HACL, compared to 0.45-0.54 from any error-state to HMD (Figure 9). Furthermore, the probability the probability of transitioning to another form of help, either HMD or HES, after HACL, was found to be 0.39 on average. In contrast, the probability to submit a correct solution in the subsequent attempt following HACL was only 0.14, compared to 0.29 and 0.24 after HES and HMD, respectively. These findings indicate that the hint was less frequently requested and had a smaller impact on progress towards solving the assignments.

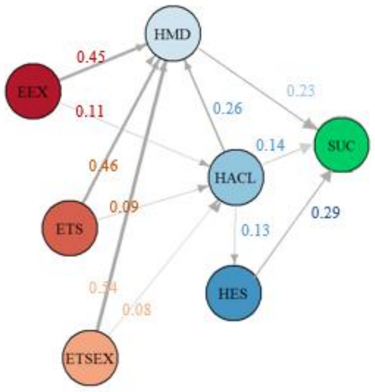


Figure 9. Transition probabilities to and from HACL, indicating low demand and low contribution.

6. DISCUSSION

In this empirical study, we implemented sequence pattern analysis to investigate the effects of automated feedback characteristics on the behavior patterns of learners, throughout solving code assignments within a MOOC for programming. For this purpose, five versions of ATF system were constructed, based on an initial version and distinctly differentiated by feedback function, message formulation, and help forms provided. Our analytic approach involved composing an “alphabet” of actions taken by learners, building learning paths and applying sequential pattern mining. Statistical tests were conducted to compare experimental groups.

Exploring learning paths revealed that integrating additional help resources to the feedback, in the form of hints and example solution, led to learners being more engaged in solving assignments by utilizing these helps. However, most learners in all the experimental groups persisted in submission attempts until arriving at the correct solution, leading to a “ceiling effect” on the achieved scores (which happens when a large percentage of observations score near the maximum grade on an assignment [28]). Here, the MOOC learning environment, characterized by unlimited attempts and the absence of knowledge evaluations, precludes an assessment of the impact of feedback structure on the level of knowledge acquisition.

The analysis did not reveal any significant impact of the motivational feedback, as the behavior of the learners in V-Motiv group was not distinct from that of learners provided with the enriched cognitive feedback (V-EC). As proposed by [27], motivational feedback may potentially have additional effects, such as on attitude and interest, but we did not collect data on these factors in the our study. Thus, among the findings of the current research, we would like to highlight learning behavior patterns exhibited by the group provided with metacognitive version (V-MC) and the group provided with the example solution version (V-ES), as the impact of these versions of feedback worth further investigation.

The metacognitive feedback is functioned to support learning through knowledge of learning strategies. In our design, the metacognitive version incorporates the strategies of help seeking and further learning through an example solution. Results suggest that to some extent, learners adopted both of these strategies. The encouragement to seek for more details about the error (HMD option) seems to affect V-MC learners towards utilizing it more often, compared to the other groups provided with the same help forms (i.e. V-EC and V-Motiv). It is worth noting that the instructions for using the ATF system, which were read by all learners *prior to starting*, include a general statement about the benefits of HMD. Thus, it is suggested the observed impact was made by the feedback, given during problem solving. Our findings support previous study, which identified a similar impact of metacognitive feedback on help-seeking behaviors, even with a long-term effect [41]. The importance of effective help-seeking in MOOCs and its association with better performance [26] highlight the potential positive contribution of metacognitive feedback to the learning process.

In addition, over half of the learners in the V-MC group utilized the example solution after submitting a correct code. This strategy has the potential to be an effective way of learning, as previously established by research [17]. [36] observed a similar engagement of students in an online course with solutions to code assignments they had already completed. Nevertheless, the motivation behind this behavior was based on prompts provided by the instructors. The ability to motivate learners in MOOCs to engage in “extra-learning“ strategy through automated feedback, without requiring instructor involvement or summative assessment, is promising.

The learners in the group provided with example-solution version (V-ES) exhibited the most distinctive learning behavior. As expected, a majority of the learners in this group utilized the provided example solution, thereby reducing the time spent on assignments. Although they have made use of the other forms of help, it was to a lesser extent in comparison to the other groups. This finding is consistent with previous research, which has determined that a solution example is perceived by students as valuable, even more so than alternative forms of feedback [38].

However, the use of the provided example did not result in "help abuse" and did not discourage learners from attempting to independently tackle code writing. The learning path of most learners indicates efforts to solve the assignments (by utilizing other forms of help) **before** resorting to opening the solution. This behavior pattern is desirable, as research suggests that novice learners may benefit from actively engaging in solution attempts before they can make sense of given example [42]. Additionally, **after** utilizing the example, many learners did not demonstrate a pattern of copy-paste the solution, but continued to attempt the assignment, although fewer submissions were necessary to arrive at the correct solution. Nevertheless, providing the entire solution for a given assignment as a feedback form is an understudied area, in contrast to step-by-step examples (e.g., [49], [47]). Further research is necessary to examine its effectiveness and to better understand learners’ perception of this type of feedback within the MOOC context.

The sequence analysis methods we applied facilitated a thorough examination of learners' utilization of the various help forms. The results indicate that the HMD was the predominant form of help, even when other forms of help such as hints or example solution were available. This finding suggests that KCR feedback, which includes only information about correctness and expected results, was not satisfying. Instead, learners sought for supplementary information. Previous studies support this finding by reporting of higher satisfaction exhibited by learners when provided with more detailed feedback for code errors [14]. On the other hand, [20] did not detect different attitudes of learners towards elaborated feedback, despite its demonstrated impact in improving performance. Additionally, unlike the study conducted by [48], our results do not provide any evidence of a connections between the use of elaborated feedback and increased engagement, as measured by the amount of time spent on completing assignments.

In line with the study of [20], hints were found as less prevalent form of help, as well as less effective, in comparison to HMD and example solution. One possible explanation is that the feedback consists of HMD is adaptive and tailored to the specific error detected, while hints in the implemented ATF are fixed and do not vary according to the submitted code. As a result, the feedback comprises HMD is geared towards rectifying identified errors, for the purpose of debugging, while the hints direct more to inquiring knowledge of concepts and found to be less useful by learners. Studies of learning environments with data-driven hints have showed different results, suggesting the use of hints shorten the time learners spent on task while achieving the same performance [38]. However, systems that include this type of support are more complex and creating the hints may be time-consuming for instructors [21]. Further study of the impact of hints' adaptivity on the extent of usage and effectiveness within MOOCs for programming is required to justify the investment of effort.

Our research approach allowed us to explore the learning behavior of students at a high resolution, identify patterns that were not observed with other tools, and compare between the experimental groups. However, we faced some challenges in utilizing this method. One of them was the significant computational time required to run multiple functions, such as searching for the most frequent sequences in each group, due to the relatively large number of learners and actions per sequence.

7. LIMITATIONS

Some limitations of this study need to be considered. First, data were collected in an authentic learning environment with low control on research setting and no indications of learners' behavior outside the course environment. In particular, other interpreters or automated feedback tools may be utilized to solve code assignments. The random assignment of learners to research groups may ensure equal tendency towards the use of such external tools, however, it has not been empirically validated.

Another limitation of this study is the narrow scope of the data analysis, which is restricted to four assignments that possess specific characteristics in terms of difficulty level and learning context. Feedback effects may vary with assignment features [32], therefore, generalizing the results of this study to diverse types of assignments should be approached with caution. Similarly, it would be beneficial to consider the features of the ATF system, such as the user interface and the manner in which various forms of help are presented (e.g. location on the screen or colors), as these factors may also have an impact on learners' interaction with feedback.

The method of SPM applied in this study had several shortcomings. Our approach involves predefining sequences of activities that represent behavioral patterns, and then analyzing their frequencies for each group. However, this method may not capture all significant differentiating patterns. An alternative approach, such as automatically capturing learning patterns from learners' interaction sequences, could potentially yield more informative findings that are relevant to the research questions. Additionally, the SPM method does not support the identification of start-to-end paths and thus the comparison between the experimental groups in terms of the entire process of solving the assignments was not allowed. A possible way to address the gap is to combine our method with process mining techniques, specifically Local Process Mining, suggested by [9], which may prove to be more compatible in the context of a single or group of assignments.

8. CONCLUSIONS AND FUTURE RESEARCH

The comparison of feedback versions in this empirical study adds to the research literature knowledge about the impact of different feedback characteristics, specifically in the context of MOOCs for programming. Significant results for our opinion, relevant to learning in MOOCs are (1) the possibility to influence learning strategies through targeted feedback function and (2) the indication for the deliberate use of example solutions by learners without negatively affecting their motivation to practice writing code themselves. These findings have implications for instructors in MOOCs, as they can use these insights to adjust the feedback provided in ATF systems to enhance support for MOOC learners. For example, to effectively encourage the use of additional help-seeking strategies such as consulting a discussion forum, or to provide additional examples of isomorphic assignments.

The data-driven approach can mitigate the gap of remote teaching and facilitate a process of on-going revising the automated feedback, by assessing the impact of changes and add-ons. Furthermore, instructors may detect problems within the assignments, by identifying, for example, patterns of repeated errors or high rates of waiving, and take steps to address these issues.

In conclusion, this study was focused on investigating the impact of feedback on learners' performance within the context of code assignments. A potential avenue for future research is to expand on this analysis and gain a more comprehensive understanding of the relationship between feedback characteristics and learning behavior. Such research may incorporate a combination of sequence and process mining methods to examine the entirety of the learners' engagement within the MOOC, including their interactions with course content (e.g., videos and comprehension exercises), through a compatible framework as previously proposed by [16].

With the increasing demand for programming skills in today's job market, MOOCs for programming have become an important tool for individuals looking to advance their careers or gain new ones, providing equal opportunities for programming education to a diverse audience. In addition, MOOCs are a valuable pedagogical supplement for instructors who seek to enhance their curriculum or provide supplementary resources for their students. We posit that this study, along with additional data-driven research in this domain, has the potential to foster the development of efficient ATF systems that promote learning in programming MOOCs, and consequently, enhance the success rate of a larger number of learners in these courses.

9. ACKNOWLEDGMENT

We express our gratitude to the Azrieli Foundation for awarding us a generous Azrieli Fellowship, which facilitated this research. Furthermore, we acknowledge our anonymous reviewers and the EDM program chairs for providing us with valuable feedback.

10. REFERENCES

[1] Ahmed, U.Z., Srivastava, N., Sindhgatta, R. and Karkare, A. 2020. Characterizing the pedagogical benefits of adaptive feedback for compilation errors by novice programmers. *Proceedings - International Conference on Software Engineering* (2020), 139–150.

[2] Benotti, L., Aloi, F., Bulgarelli, F. and Gomez, M.J. 2018. The effect of a web-based coding tool with automatic feedback on students' performance and perceptions. *SIGCSE 2018 - Proceedings of the 49th ACM Technical Symposium on Computer Science Education* (2018), 2–7.

[3] Bogarín, A., Cerezo, R. and Romero, C. 2018. A survey on educational process mining. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*. 8, 1 (Jan. 2018),

e1230. DOI= https://doi.org/10.1002/WIDM.1230.

[4] Boroujeni, M.S. and Dillenbourg, P. 2018. Discovery and temporal analysis of latent study patterns in MOOC interaction sequences. *ACM International Conference Proceeding Series* (Mar. 2018), 206–215.

[5] Cavalcanti, A.P., Barbosa, A., Carvalho, R., Freitas, F., Tsai, Y.-S., Gašević, D. and Mello, R.F. 2021. Automatic feedback in online learning environments: A systematic literature review. *Computers and Education: Artificial Intelligence*. 2, (Jan. 2021), 100027. DOI= https://doi.org/10.1016/J.CAEAI.2021.100027.

[6] Chan, Y. and Walmsley, R.P. 1997. Learning and understanding the Kruskal-Wallis one-way analysis-of-variance-by-ranks test for differences among three or more independent groups. *Physical Therapy*. 77, 12 (1997), 1755–1762. DOI= https://doi.org/10.1093/ptj/77.12.1755.

[7] Deeva, G., Bogdanova, D., Serral, E., Snoeck, M. and De Weerdt, J. 2021. A review of automated feedback systems for learners: Classification framework, challenges and opportunities. *Computers & Education*. 162, (Mar. 2021), 104094. DOI= https://doi.org/10.1016/J.COMPEDU.2020.104094.

[8] Deeva, G., De Smedt, J., De Koninck, P. and De Weerdt, J. 2018. Dropout prediction in MOOCs: A comparison between process and sequence mining. *Lecture Notes in Business Information Processing*. 308, January (2018), 243–255. DOI= https://doi.org/10.1007/978-3-319-74030-0_18.

[9] Deeva, G. and Weerdt, J. De 2018. Understanding Automated Feedback in Learning Processes by Mining Local Patterns. *Lecture Notes in Business Information Processing*. 342, (Sep. 2018), 56–68. DOI= https://doi.org/10.1007/978-3-030-11641-5_5.

[10] Derval, G., Gego, A., Reinbold, P., Benjamin, F. and Van Roy, P. 2015. Automatic grading of programming exercises in a MOOC using the INGInious platform.. *Proceedings of the European Stakeholder Summit on experiences and best practices in and around MOOCs (EMOOCS'15)* (Mons, Belgium, 2015), 91–86.

[11] Fan, Y., Tan, Y., Rakovi, M., Wang, Y., Cai, Z., Williamson Shaffer, D., Gaševi, D. and Yizhou Fan, C. 2022. Dissecting learning tactics in MOOC using ordered network analysis. *Journal of Computer Assisted Learning*. 39, 1 (Aug. 2022), 154–166. DOI= https://doi.org/10.1111/JCAL.12735.

[12] Faucon, L., Kidzí, Ł. and Dillenbourg, P. 2016. Semi-Markov model for simulating MOOC students. *Proceedings of the 9th International Conference on Educational Data Mining, EDM 2016* (2016), 358–363.

[13] Félix, E., Amadieu, F., Venant, R. and Broisin, J. 2022. Process and Self-regulation Explainable Feedback for Novice Programmers Appears Ineffectual. *Proceedings of the European Conference on Technology Enhanced Learning* (2022), 514–520.

[14] Finn, B., Thomas, R. and Rawson, K.A. 2018. Learning more from feedback: Elaborating feedback with examples enhances concept learning. *Learning and Instruction*. 54, (Apr. 2018), 104–113. DOI= https://doi.org/10.1016/j.learninstruc.2017.08.007.

[15] Gabadinho, A., Ritschard, G., Müller, N.S. and Studer, M. 2011. Analyzing and Visualizing State Se-quences in R with TraMineR. *Journal of Statistical Software*. 40, 4 (2011), 1–37. DOI= https://doi.org/10.18637/jss.v040.i04.

[16] Gabbay, H. and Cohen, A. 2022. Investigating the effect of automated feedback on learning behavior in MOOCs for programming. *EDM 2022 - Proceedings of the 15th International Conference on Educational Data Mining* (2022), 376–383.

[17] Garcia, R., Falkner, K. and Vivian, R. 2018. Systematic literature review: Self-Regulated Learning strategies using e-learning tools for Computer Science. *Computers and Education*. 123, (Aug. 2018), 150–163. DOI= https://doi.org/10.1016/j.compedu.2018.05.006.

[18] Gross, S. and Pinkwart, N. 2015. How do learners behave in help-seeking when given a choice? *Artificial Intelligence in Education: 17th International Conference, AIED 2015* (Madrid, Spain, 2015), 600–603.

[19] Hao, Q., Smith IV, D.H., Ding, L., Ko, A., Ottaway, C., Wilson, J., Arakawa, K.H., Turcan, A., Poehlman, T. and Greer, T. 2022. Towards understanding the effective design of automated formative feedback for programming assignments. *Computer Science Education*. 32, 1 (Jan. 2022), 105–127. DOI= https://doi.org/10.1080/08993408.2020.1860408.

[20] Hao, Q., Wilson, J.P., Ottaway, C., Iriumi, N., Arakawa, K. and Smith, D.H. 2019. Investigating the essential of meaningful automated formative feedback for programming assignments. *Proceedings of IEEE Symposium on Visual Languages and Human-Centric Computing, VL/HCC* (Oct. 2019), 151–155.

[21] Keuning, H., Jeuring, J. and Heeren, B. 2018. A systematic literature review of automated feedback generation for programming exercises. *ACM Transactions on Computing Education*. 19, 1 (2018), 1–43. DOI= https://doi.org/10.1145/3231711.

[22] Kiesler, N. 2022. An Exploratory Analysis of Feedback Types Used in Online Coding Exercises. *arXiv preprint* . 2022)).

[23] Kinnebrew, J.S., Loretz, K.M. and Biswas, G. 2013. A Contextualized, Differential Sequence Mining Method to Derive Students' Learning Behavior Patterns. *Journal of Educational Data Mining*. 5, 1 (2013), 190–219.

[24] Li, S., Du, J. and Sun, J. 2022. Unfolding the learning behaviour patterns of MOOC learners with different levels of achievement. *International Journal of Educational Technology in Higher Education*. 19, 1 (2022), 1–20. DOI= https://doi.org/10.1186/s41239-022-00328-8.

[25] Maldonado-Mahauad, J., Pérez-Sanagustín, M., Kizilcec, R.F., Morales, N. and Munoz-Gama, J. 2018. Mining theory-based patterns from Big data: Identifying self-regulated learning strategies in Massive Open Online Courses. *Computers in Human Behavior*. 80, (Mar. 2018), 179–196. DOI= https://doi.org/10.1016/j.chb.2017.11.011.

[26] Maldonado-Mahauad, J., Pérez-Sanagustín, M., Moreno-Marcos, P.M., Alario-Hoyos, C., Muñoz-Merino, P.J. and Delgado-Kloos, C. 2018. Predicting Learners' Success in a Self-paced MOOC Through Sequence Patterns of Self-regulated Learning. *Lifelong Technology-Enhanced Learning: 13th European Conference on Technology Enhanced Learning, EC-TEL 2018* (Leeds, UK, 2018), 355–369.

[27] Marwan, S., Fisk, S., Price, T.W., Barnes, T. and Gao, G. 2020. Adaptive Immediate Feedback Can Improve Novice Programming Engagement and Intention to Persist in Computer Science. *Proceedings of the 2020 ACM Conference on International Computing Education Research* (2020), 194–203.

[28] Marwan, S. and Price, T.W. 2022. iSnap : Evolution and Evaluation of a Data-Driven Hint System for Block-based Programming. XX, X (2022), 1–15. DOI= https://doi.org/10.1109/TLT.2022.3223577.

[29] Marwan, S., Williams, J.J. and Price, T. 2019. An evaluation of the impact of automated programming hints on performance and learning. *ICER 2019 - Proceedings of the 2019 ACM Conference on International Computing Education Research*. (Jul. 2019), 61–70. DOI= https://doi.org/10.1145/3291279.3339420.

[30] McBroom, J., Yacef, K., Koprinska, I. and Curran, J.R. 2018. A data-driven method for helping teachers improve feedback in computer programming automated tutors. *Artificial Intelligence in Education: 19th International Conference, AIED 2018, Proceedings, Part I* (London, UK, 2018), 324–337.

[31] Mitrovic, A., Ohlsson, S. and Barrow, D.K. 2013. The effect of positive feedback in a constraint-based intelligent tutoring system. *Computers and Education*. 60, 1 (Jan. 2013), 264–272. DOI= https://doi.org/10.1016/j.compedu.2012.07.002.

[32] Narciss, S. 2013. Designing and evaluating tutoring feedback strategies for digital learning environments on the basis of the interactive tutoring feedback model. *Digital Education Review*. 23, 1 (2013), 7–26. DOI= https://doi.org/10.1344/der.2013.23.7-26.

[33] Narciss, S. 2008. Feedback strategies for interactive learning tasks. *Handbook of research on educational communications and technology*. J.M. Spector, M.D. Merrill, J. Van Merriënboer, and M.P. Driscoll, eds. Taylor and Francis. 125–143.

[34] Perikos, I., Grivokostopoulou, F. and Hatzilygeroudis, I. 2017. Assistance and Feedback Mechanism in an Intelligent Tutoring System for Teaching Conversion of Natural Language into Logic. *International Journal of Artificial Intelligence in Education 2017 27:3*. 27, 3 (Feb. 2017), 475–514. DOI= https://doi.org/10.1007/S40593-017-0139-Y.

[35] Pieterse, V. 2013. Automated Assessment of Programming Assignments. *3rd Computer Science Education Research Conference on Computer Science Education Research*. 3, April (2013), 45–56.

[36] Price, T.W., Williams, J.J., Solyst, J. and Marwan, S. 2020. Engaging Students with Instructor Solutions in Online Programming Homework. *Conference on Human Factors in Computing Systems - Proceedings* (Apr. 2020), 1–7.

[37] Rajendran, R., Iyer, S. and Murthy, S. 2019. Personalized Affective Feedback to Address Students' Frustration in ITS. *IEEE Transactions on Learning Technologies*. 12, 1 (Jan. 2019), 87–97. DOI= https://doi.org/10.1109/TLT.2018.2807447.

[38] Rivers, K. 2018. Automated data-driven hint generation for learning programming. *Dissertation Abstracts International: Section B: The Sciences and Engineering*. 79, 4-B(E) (2018).

[39] Rizvi, S., Rienties, B., Rogaten, J. and Kizilcec, R.F. 2020. Investigating variation in learning processes in a FutureLearn MOOC. *Journal of Computing in Higher Education*. 32, (2020), 162–181. DOI= https://doi.org/10.1007/s12528-019-09231-0.

[40] Rohani, N., Gal, K., Gallagher, M. and Manataki, A. 2023. Discovering Students ' Learning Strategies in a Visual Programming MOOC through Process Mining Techniques. *Process Mining Workshops: ICPM 2022 International Workshops, Revised Selected Papers* (Bozen-Bolzano, Italy, 2023), 539–551.

[41] Roll, I., Aleven, V., McLaren, B.M. and Koedinger, K.R. 2011. Improving students' help-seeking skills using metacognitive feedback in an intelligent tutoring system. *Learning and Instruction*. 21, 2 (Apr. 2011), 267–280. DOI= https://doi.org/10.1016/j.learninstruc.2010.07.004.

[42] Roll, I., Baker, R.S.J. d., Aleven, V. and Koedinger, K.R. 2014. On the Benefits of Seeking (and Avoiding) Help in Online Problem-Solving Environments. *Journal of the Learning Sciences*. 23, 4 (Oct. 2014), 537–560. DOI= https://doi.org/10.1080/10508406.2014.883977.

[43] Saint, J., Gaševic, D., Matcha, W., Uzir, N.A.A. and Pardo, A. 2020. Combining analytic methods to unlock sequential and temporal patterns of self-regulated learning. *ACM International Conference Proceeding Series*. (Mar. 2020), 402–411. DOI= https://doi.org/10.1145/3375462.3375487.

[44] Serth, S., Teusner, R. and Meinel, C. 2021. Impact of Contextual Tips for Auto-Gradable Programming Exercises in MOOCs. *Proceedings of the Eighth ACM Conference on Learning @ Scale* (New York, NY, USA, 2021), 307–310.

[45] Shabrina, P., Marwan, S., Chi, M., Price, T.W. and Barnes, T. 2020. The Impact of Data-driven Positive Programming Feedback: When it Helps, What Happens when it Goes Wrong, and How Students Respond. *Educational Data Mining in Computer Science Education Workshop@ EDM 2020* (2020).

[46] Vizcaíno, A. 2005. A Simulated Student Can Improve Collaborative Learning. *International Journal of Artificial Intelligence in Education*. 15, (2005), 3–40.

[47] Wang, W., Rao, Y., Zhi, R., Marwan, S., Gao, G. and Price, T.W. 2020. Step Tutor: Supporting Students through Step-by-Step Example-Based Feedback. *Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE*. (2020), 391–397. DOI= https://doi.org/10.1145/3341525.3387411.

[48] Wang, Z., Gong, S.Y., Xu, S. and Hu, X.E. 2019. Elaborated feedback and learning: Examining cognitive and motivational influences. *Computers and Education*. 136, (Jul. 2019), 130–140. DOI= https://doi.org/10.1016/j.compedu.2019.04.003.

[49] Zhi, R., Marwan, S., Dong, Y., Lytle, N., Price, T.W. and Barnes, T. 2019. Toward data-driven example feedback for novice programming. *EDM 2019 - Proceedings of the 12th International Conference on Educational Data Mining* (2019), 218–227.

[50] Zhou, Y., Andres-Bray, J.M., Hutt, S., Ostrow, K. and Baker, R.S. 2021. A Comparison of Hints vs. Scaffolding in a MOOC with Adult Learners. *Artificial Intelligence in Education: 22nd International Conference, AIED 2021, Proceedings, Part II* (Utrecht, The Netherlands, Jun. 2021), 427–432.