Kattis vs ChatGPT: Assessment and Evaluation of Programming Tasks in the Age of Artificial Intelligence

Dunder, N.
ndunder@kth.se
KTH Royal Institute of Technology
Stockholm, Sweden

Lundborg, S.
sagalu@kth.se
KTH Royal Institute of Technology
Stockholm, Sweden

Wong, J.
l.y.j.wong@uu.nl
Utrecht University
Hekla, Netherlands

Viberg, O.
oviberg@kth.se
KTH Royal Institute of Technology
Stockholm, Sweden

ABSTRACT

AI-powered education technologies can support students and teachers in computer science education. However, with the recent developments in generative AI, and especially the increasingly emerging popularity of ChatGPT, the effectiveness of using large language models for solving programming tasks has been underexplored. The present study examines ChatGPT's ability to generate code solutions at different difficulty levels for introductory programming courses. We conducted an experiment where ChatGPT was tested on 127 randomly selected programming problems provided by *Kattis*, an automatic software grading tool for computer science programs, often used in higher education. The results showed that ChatGPT independently could solve 19 out of 127 programming tasks generated and assessed by *Kattis*. Further, ChatGPT was found to be able to generate accurate code solutions for simple problems but encountered difficulties with more complex programming tasks. The results contribute to the ongoing debate on the utility of AI-powered tools in programming education.

CCS CONCEPTS

• **Applied computing** → **Education**; • **Computing methodologies** → **Artificial intelligence**; • **Hardware** → **Emerging technologies**.

KEYWORDS

Programming Education, ChatGPT, Automated Grading, Academic Integrity

ACM Reference Format:
Dunder, N., Lundborg, S., Wong, J., and Viberg, O.. 2024. Kattis vs ChatGPT: Assessment and Evaluation of Programming Tasks in the Age of Artificial Intelligence. In *The 14th Learning Analytics and Knowledge Conference (LAK '24), March 18–22, 2024, Kyoto, Japan.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3636555.3636882

LAK '24, March 18–22, 2024, Kyoto, Japan

ACM ISBN 979-8-4007-1618-8/24/03
https://doi.org/10.1145/3636555.3636882

1 INTRODUCTION

Programming education is "an important source of skills and knowledge for students and a necessary feature to survive in a competitive job market" ([32], p.2). During the last decade, artificial intelligence(AI)-powered education technologies have been used in programming education to assist teachers in guiding students to acquire programming and computational skills and knowledge [12]. Such AI-powered technologies include various types of intelligent tutoring systems (for an overview, see [6]), plagiarism detection tools [5] and auto-grading tools such as *Kattis* (e.g., [2]). Therefore, AI is not futuristic but actively used in classrooms and courses worldwide [11].

Recently, new kinds of AI-powered tools, namely generative AI technologies - "a distinct class of AI and an incredibly powerful technology that has been popularized by ChatGPT" ([19], p.2) - have rapidly penetrated different parts of our society, including learning and teaching practices in the setting of programming higher education [12, 18, 29, 32]. Whereas earlier research on using AI-powered education tools in programming education has shown related evidence in terms of their efficacy to support students in coding tasks by, for example, providing suggestions [26], error detection [14], and automatic code generation [2], our understanding of ChatGPT and other rapidly emerging large language models' effectiveness for accurately solving coding tasks is still limited [18, 29]. Knowing the capabilities of these tools is important since they are already extensively used by students and can generate the code from the description, perform code completion, translation, and summarization [20].

Evidence for the power and potential to solve programming tasks using ChatGPT in introductory-level programming education is emerging [12]. All this offers several new opportunities to improve students' conditions of learning. However, at the same time, it raises several questions and concerns, including those that pertain to the degree of accuracy of the provided solutions and the student's intention to use such tools to complete course assignments. Concerns were raised by [3], given that it has become much easier for students to produce codes to pass traditional first-year programming assignments and even exams using generative AI-powered tools. This brings us to a heavily debated concern about academic integrity in higher education in the era of generative AI and, on the other hand, the opportunities for enhancing learning and teaching [28].

Central to these concerns is a larger question about ChatGPT's capacity to empower students in programming education. As stressed by Steele [27], AI chatbots such as ChatGPT can threaten contemporary education systems in terms of: "(1) measurement of students' knowledge and skills; (2) accuracy of the information students are learning; and (3) the market value of the skills we are teaching" (p.1). Addressing such concerns requires examining the kinds of tasks AI chatbots can perform, and how well they can perform. Answers to these questions will offer us an evidence-based ground to consider the degree to which ChatGPT could be effectively integrated into programming education to improve students' conditions for learning.

The present study aims to evaluate the extent to which ChatGPT-3.5 can accurately solve programming tasks (at varied difficulty levels) provided and corrected by an automated code generation and assessment tool, *Kattis*, which is widely used in introductory programming courses of engineering education at several highly ranked higher educational institutions across countries. The focus on the introductory level of programming education, in which the present study has been conducted, is important since, at this level, students need extensive practice in writing codes, among other activities, to gain fundamental programming skills [30]. Our study contributes to building a better understanding of ChatGPT's capabilities to inform teaching and learning practices by answering our main research question: *To what extent is ChatGPT able to solve automatically generated coding tasks in the setting of introductory programming education?*

2 BACKGROUND

2.1 Automatic assessment of programming tasks

The development of coding assignments and their assessment in programming education is a highly time-consuming task for educators [9], especially in the context of large courses (i.e.,>100 students participate in one course), as in the case of the present study. During the last few years, and especially from 2021, such task has been supported by AI-driven code generation tools such as *OpenAI Codex* and *Amazon CodeWhispeer*, which are argued to be able to support students and educators in their everyday educational practices [3, 17]. Such tools can be used to support educators by automatically generating programming exercises at various levels of difficulty and by generating guiding hints to coding solutions, like in the case of *Kattis*. OpenAI and DeepMind have recently introduced groundbreaking generative AI-models such as ChatGPT-3.5 that are capable of not only generating coding assignments but also - computer code, potentially making programming more productive and accessible [3]. Such tools have hitherto been freely accessible to students, suggesting that some students participating in programming education are using AI code compilation in their coding assignments.

In the current study, we examine ChatGPT-3.5's solutions to programming tasks provided and assessed by *Kattis*. *Kattis*, available freely online (open.kattis.com), is an automated code generation and grading system introduced in programming courses at a large European university in 2002 and has since gained widespread popularity in higher education. Its primary purpose is not only to generate programming tasks automatically but also to shift the responsibility of assessing the correctness of program code from the instructor to an automated tool, thereby releasing the instructor's time that can instead be used to assist students continuously during lab sessions and other related tasks. Additionally, *Kattis* serves as an online judge for submissions in programming competitions [8].

2.2 ChatGPT in higher education

ChatGPT is most popularly used in higher education settings compared to K-12 education and training of practical skills [13]. As stressed by [7], students' self-initiated adoption of ChatGPT has made it almost impossible to ban or control it. Its rapid uptake rate among students has induced a student-driven educational tool" (p.86). The release and rapid diffusion of ChatGPT have caught educators' attention worldwide due to its and other AI-based technologies to "transform education" ([10], p.1).

On the one hand, early research exploring teachers' attitudes toward using ChatGPT has shown that educators are generally cautious in their approach to using ChatGPT [15]. On the other hand, the results of recent studies have shown that students – another key stakeholder of generative AI tools in education – overall demonstrate positive attitudes toward ChatGPT but raised concerns about privacy, ethical issues, the impact on personal development and career prospects [4]. A recent review [1] of 14 empirical studies showed that ChatGPT was beneficial for learning and teaching. Concerning learning, the reviewed studies showed that students used ChatGPT in various ways: as an intelligent assistant for answering on-demand questions, searching for information, and receiving feedback. However, the findings on students' perceived accuracy, relevance, and reliability of ChatGPT's output are mixed.

Among early results summarizing the use of ChatGPT in higher education (N = 12 papers examined), scholars highlight that the implementation of ChatGPT in education has a positive influence on the teaching and learning process but also stress the importance of teachers being trained to use the tool effectively [22]. In programming education, teachers and students have similarly indicated that generative AI-powered tools would play a significant role. They also stressed several concerns about how large language models should be best integrated to support their needs [33]. The examination of ChatGPT's performance against AI-powered EdTech tools used in higher education for some time is so far limited. While some evidence on ChatGPT is emerging, it is still in its infancy due to the lack of rigorous evaluations of the impact of the use of ChatGPT on learning and teaching. Evaluating ChatGPT's capabilities, for example, in solving programming tasks, is needed to make well-informed teaching, assessment, and evaluation decisions.

2.3 ChatGPT in programming education

Programming skills and knowledge are becoming increasingly important in today's world which is rapidly evolving with the acceleration of technological and digital advancement [32]. This study, in particular, focuses on *introductory* programming education since it is considered to be especially challenging for students due to their inability to comprehend what is happening to their

program in memory because they are incapable of creating a clear mental model of its execution [21]. With the emergence of ChatGPT in programming education, scholars argue that "generative AI-powered tools can transform programming education" ([32], p.2) by using it as a bot for discussing source code and even generating code [29], among others.

Whereas ChatGPT could be used in several ways in programming education, little is known about its performance in solving programming problems in introductory programming education. Geng and colleagues [12] have explored how well ChatGPT (treated as one of the students) can perform in an introductory-level functional language programming course and found that it can achieve a grade *B*, thus successfully passing the course. At the same time, the authors stress the importance of using ChatGPT in tandem with other teaching methods to ensure that students develop a well-rounded set of programming skills [12]. Another recent study investigated the performance of ChatGPT-3.5 and GPT-4 in solving programming tasks [18]. The results of evaluating 72 Python tasks' solutions (retrieved from the open source platform *CodingBat*) for novice programmers were compared to the ChatGPT's performance. The findings demonstrated high scores of 94.4 to 95.8 percent correct responses. However, the authors stress that model solutions to all *CodingBat* tasks are available in GitHub, suggesting that the chances that ChatGPT was trained in such data are high. Finally, [25] evaluated the capability of ChatGPT to pass assessments in introductory and intermediate Python programming courses. The results showed that the current models are not capable of passing the full spectrum of assessments included in a Python programming course (<%70 on even entry-level modules). All in all, the related evidence is scarce and inconsistent, suggesting that more empirical research is needed.

3 METHOD

3.1 Study Design

The present study was conducted in the setting of a large introductory programming course (i.e., > 100 students) that is a mandatory part of several engineering programs at a large technical university in Europe. *Kattis*, an automated code-generation and grading tool has been used in the targeted setting of programming education for several years, and was found to be efficient in supporting teachers and students [2]. Coding tasks from *Kattis* were sent to Open AI's freely accessible large language model, ChaptGPT-3.5, to generate solutions in Python code. The assessment results generated by *Kattis* were considered as the dependent variable, and the solutions provided by ChatGPT as the independent variable. For examples of coding tasks, solutions, and metadata, as well as a clarification of the Kattis rating process, please refer to this link: https://osf.io/p8xz6.

3.2 Coding Tasks

The coding tasks (N = 127) were randomly selected from *Kattis* in spring 2023, with consideration for an even distribution across difficulty levels. Figure 1 illustrates the distribution of coding tasks across several difficulty levels. Each task in *Kattis* comes with metadata, indicating the difficulty level of the task, ranging from 1 to 10, and further categorized as 'easy' (1.0-2.7), 'medium' (2.8-5.3), and 'hard' (5.4-10.0). The difficulty level of a coding task is estimated by using a variant of the Elo rating system (see e.g., [24]). Specifically, whereas the tasks that have been solved by many people with only a few attempts indicate a lower degree of difficulty, the tasks that have been attempted to be solved by many individuals but rarely solved indicate a higher difficulty level. Tasks that have very few submissions tend to fall under the category of 'medium' difficulty as *Kattis* lacks sufficient data on their difficulty level (https://open.kattis.com/help/ranklist). In this study, difficulty levels of the coding tasks were re-categorized as integers from 1 to 10 for clearer result visualization. That is, all tasks with a difficulty level between 1.0 to 1.9 are classified as tasks of difficulty Level 1. Further, those tasks that correspond to difficulty Levels 1 and 2 are grouped within a larger category of 'easy. Difficulty levels of 3 and 4 are grouped as 'medium', and difficulty levels from 5 to 10 were grouped as 'hard'. All the coding tasks were copied directly from *Kattis*. One constraint relates to the fact that the choice of the coding tasks was limited to those that were possible to copy and transfer to ChatGPT easily; the tasks with a large number of mathematical equations and figures (common to tasks with high levels of difficulty) were excluded due to the technical limitations.

Many of the coding tasks require solutions consisting of code ranging from 3 to 20 lines, with an average of around 10 lines. Most of the tasks can be solved by correctly formatting the input data and then performing a task. In many tasks, data is formatted into lists and dictionaries, sometimes tuples or sets. The operation performed is often either a for loop, a while loop, a mathematical operation, or sorting, possibly followed by an 'if-else' statement where the result is printed.

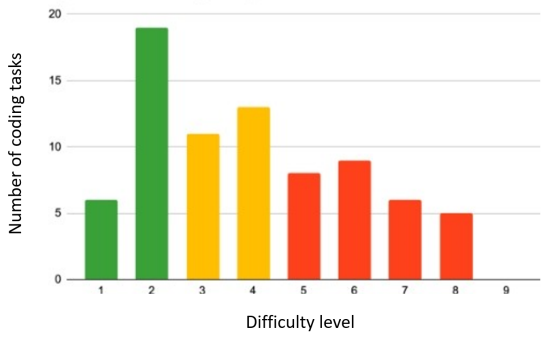

Figure 1: Distribution of the number of coding tasks across the difficulty levels

Apart from the difficulty level, each coding task in *Kattis* comes with metadata. Table 1 provides a description of the metadata that was retrieved from *Kattis* to examine the association with the assessment of the coding solutions provided by ChatGPT.

3.3 Study Procedure

Each coding task from *Kattis* was sent to ChatGPT to generate a solution proposal in Python code. The proposed solution was then submitted to *Kattis'* own text window, which returned a response

Table 1: Description on metadata retrieved from Kattis

Metadata	Description
Approved Submissions	% approved submissions of all submitted solutions in Kattis. Interval between 0-100.
Successful Submitters	% of users who tried and then succeeded in solving the code problem. Range between 1-100.
Difficulty Level	A number on a scale of 1-10, where 1 is the easiest and 10 is the most difficult.

as to whether the solution was approved or not. The type of error message from *Kattis* (i.e., feedback) was noted in cases where an approved verdict could not be given.

3.4 Data Analysis

To assess the performance of ChatGPT, we first determined the percentage of problems solved (i.e., solutions approved by *Kattis*). We have also examined ChatGPT's performance for the level of the task's difficulty, where there was at least one approved solution. For the solutions that were not approved by *Kattis*, the frequency and type of error messages were analyzed. To further examine the extent of ChatGPT's performance against the general performance on the coding tasks, we finally performed a correlational analysis on the number of approved solutions from ChatGPT in *Kattis* (i.e., coded as a binary variable where '1' is accepted and '0' is not accepted) and the metadata retrieved from *Kattis* (Table 1).

4 RESULTS

4.1 Performance of ChatGPT- approved solutions

Out of the 127 code solutions generated by ChatGPT, only 19 (15%) were fully approved by *Kattis*. Figure 2 illustrates the overall distribution of the coding tasks and the difficulty level. Green bars represent approved solutions and blue bars are the solutions that were not approved by *Kattis*. Among the 19 approved solutions, 10 were solutions to the coding tasks with the difficulty Level 1 (8% out of the overall sample of 127 tasks), seven corresponded to the difficulty Level 2 (6%), and only two - the difficulty Level 4 (2%). The majority (85%) of the coding solutions generated by ChatGPT were not approved by *Kattis*. The results suggest a low performance by ChatGPT in terms of both correctly solving the tasks and its ability to solve high-level difficulty tasks.

Further inspection shows that, among all the approved solutions, the approved solution for the coding task with the lowest level of difficulty is for the task with a difficulty level of 1.3, while the one with the highest level of difficulty was rated at 4.2. Given that the task description, provided by *Kattis* for the easiest task is much shorter than the task description for the most challenging task, ChatGPT also provided solutions of different lengths and quality: the easiest task consists of only 11 lines of code and mainly 'if'- and 'else'-statements, while the most challenging task is composed of 55 lines of code and include 'loops', 'functions', 'lists', and 'matrices'.

4.2 Performance of ChatGPT- failed solutions

Kattis provided feedback in the form of error messages for the 108 coding tasks with incorrect coding solutions (i.e., ChatGPT solutions that were not approved by *Kattis*). The *'Wrong Answer'* feedback occurred for 83 tasks (77% of all incorrect solutions) where *Kattis* did not approve the code solution. The second most common error was *'Run Time Error'* that occurred in 16 tasks (15%), and the least common one was *'Time Limit Exceeded'* that occurred for 9 tasks (8%). This suggests that program crashes were the most frequent issue while running for too long was the least common problem.

Out of the 108 incorrect coding solutions, 19 indicated partially accepted solutions, referring to the coding solutions that were accepted for certain inputs and not all the inputs. This accounts for 18% of failed solutions and 15% of all solutions. For each coding task,*Kattis* assessed the solution for a different number of inputs. Table 2 provides an overview of the ratio of inputs accepted by *Kattis*, the assessment (i.e., the error message shown to the submitter as feedback on the coding solution), and the difficulty level of the partially accepted coding tasks. In this set of coding tasks, *'Wrong Answer'* occurred for most of the coding tasks, followed by textit'Time Limit Exceeded' and *'Run Time Error'*.

Among the set of partially accepted solutions, the most number of inputs that were approved (i.e., n = 12) was found for two tasks, one with difficulty level 3.9 and the other with difficulty level 4.1. The coding solution that had the highest percentage of inputs accepted was of difficulty level 2.8; it was accepted for 60% of the inputs. The coding solution for the task with the highest difficulty level (i.e., 6.5) was accepted for only 1 out of 13 inputs. This suggests a varying performance of ChatGPT across tasks of different difficulty levels.

4.3 Correlations analysis: Evaluation of ChatGPT's performance against general performance

Table 3 shows the the correlational table for ChatGPT's performance in terms of the accepted solutions and the general performance indicators on the selected coding tasks, based on the metadata (for a description of the metadata, see Table 1). Positive correlations of 0.43 and 0.37 were found between approved solutions of ChatGPT and the number of submissions for the coding tasks, and between approved solutions of ChatGPT and successful 'submitters' respectively. Therefore, ChatGPT's ability to solve a coding task correlates slightly more with the approved

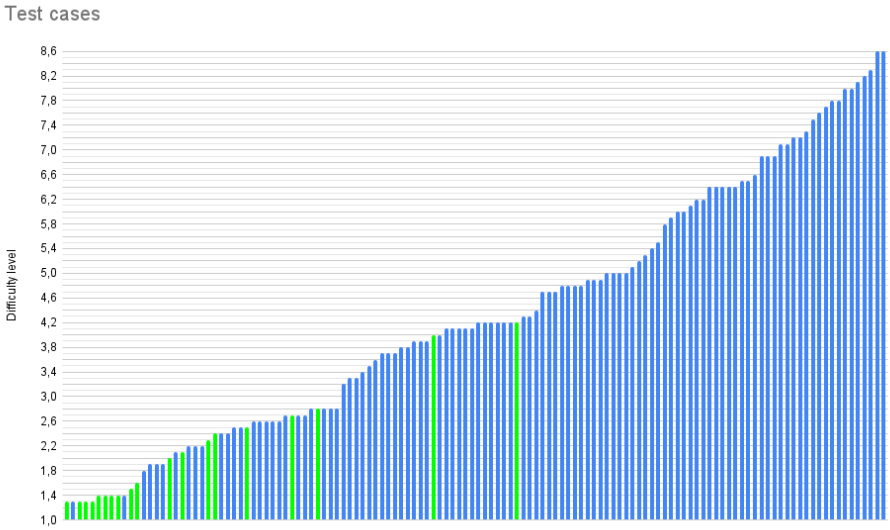

Figure 2: Distribution graph of the 127 tasks at different difficulty levels (1-10). Green bars represent tasks approved by *Kattis*, and blue bars – not approved.

Table 2: Overview of the type of errors and difficulty level of partially accepted solutions

Ratio of approved inputs	% of approval	Kattis' assessment	Difficulty level
1/13	8%	WA	1.9
3/12	25%	TLE	2.2
1/33	3%	WA	2.4
1/38	3%	WA	2.5
5/23	22%	WA	2.6
1/13	8%	WA	2.6
1/78	1%	WA	2.7
5/28	18%	WA	2.6
3/5	60%	TLE	2.8
1/34	3%	TLE	3.2
3/11	27%	WA	3.3
2/13	15%	WA	3.5
12/21	57%	WA	3.9
12/40	30%	WA	4.1
1/2	50%	RTE	4.8
2/52	4%	WA	5.0
5/18	28%	TLE	6.2
2/17	12%	TLE	6.4
1/13	8%	WA	6.5

submissions than with the number of successful submitters. The negative correlation of -0.48 between the level of difficulty and approved solutions of ChatGPT indicates that as the difficulty level of the coding tasks increases, the number of approved solutions from ChatGPT decreases. Overall, the results suggest that ChatGPT performs better on easier coding tasks with a high percentage of submissions already approved in the system.

5 DISCUSSION AND CONCLUSIONS

Considering that generative AI-powered tools can transform programming education [32], many questions remain regarding

Table 3: Correlations between ChatGPT's performance and indicators of general performance

Variable	M	SD	1	2	3	4
1. Approved Submissions	36.8	13.5	-			
2. Successful Submitters	81.8	12.4	0.71***	-		
3. Difficulty Level	4.3	2	−0.70***	−0.84***	-	
4. Approved Solutions of ChatGPT (=1)	0.1	0.4	0.43***	0.37***	−0.48***	-

* $P \leq 0.05$, ** $P \leq 0.01$, *** $P \leq 0.001$

how this transformation could be facilitated, ultimately leading to the student-improved acquisition of programming skills and knowledge. To fill this gap, this study examined the extent to which ChatGPT is capable of solving automatically generated coding tasks in the setting of introductory programming education. By examining ChatGPT's ability to solve coding tasks (at different levels of difficulty), automatically generated and assessed by *Kattis,* (i.e., the system that is frequently used in computer science education for the provision of feedback and assessment across countries), we found that the current capability of ChatGPT to solve such tasks in the targeted setting is somewhat limited (only 15% were approved by *Kattis*). These results are not supported by earlier research findings (e.g., [12, 18]), but are in line with the results of another recent study by [25]. Our results also give us a nuanced picture of the difficulty level of programming tasks that ChatGPT can solve, as shown by how it can currently solve mostly the tasks categorized as 'easy'.

In general, these findings imply that over-reliance on the code solutions provided by ChatGPT today may hamper students' acquisition of programming skills and knowledge, and can be especially detrimental for students in introductory programming education. However, this does not suggest that ChatGPT should be banned since as shown by [32], its use, as experienced by students, can improve their thinking skills, increase their self-confidence when solving programming tasks, and facilitate debugging. Instead, educators are recommended to design and integrate teaching activities that would enable students to critically reflect and use ChatGPT for learning by self-evaluating the responses (to programming tasks) provided by generative AI-powered tools such as ChatGPT or similar. To achieve this, educators need to be supported in terms of their professional development focusing on teaching with AI, since many of them may lack relevant knowledge and skills that constitute AI digital competencies (e.g., [23]).

Furthermore, since ChatGPT's use has been predominantly driven by students [7], they need to be supported in the development of relevant skills, including their critical thinking- and self-regulated learning (SRL) skills, which are positively associated with their academic performance in online and blended learning settings [31], and in which the prevalent part of introductory programming education is offered. Such support can be considered in several ways. One way is to ensure that the fostering of students' critical thinking- and SRL skills are a part of the introductory program education curriculum. Another complementary way is to explore the opportunities of how ChatGPT can be used to effectively support students in their development of such skills. As recently demonstrated by scholars, using AI applications for supporting students' SRL in online learning can be effective [16].

This study has several limitations. First, only 127 programming tasks have been tested. This limits our ability to generalize the results. Second, the tests were performed over one month on different days, and during that time, ChatGPT-3.5 was updated (March 23, 2023), which may have influenced our results. For more controlled experimental settings, we recommend performing all the tests on the same day. Third, the study assumes how students might generate code solutions from ChatGPT by copying and pasting the task descriptions provided by *Kattis*. It is not known if students might use ChatGPT in other ways, such as using prompt engineering to generate partial codes and evaluating the accuracy of the codes themselves.

To better understand the key stakeholders' perspectives on utilizing ChatGPT in teaching and learning programming skills, future research should focus on following up qualitative research studies targeting both students and teachers. Second, since new large language models are continuously developing, scholars need to continuously evaluate their capabilities in authentic settings of introductory programming education. Third, studies in which ChatGPT is assisted by a human (either the teacher or the student) in solving tasks at higher difficulty levels are recommended to gain a deeper insight into the benefits and limitations of human-AI collaboration for student-improved learning of programming skills. Finally, rigorous experimental studies that measure changes in student learning are needed to assess the impact of utilizing ChatGPT or similar chatbots on students' acquisition of programming skills and their academic performance.

In sum, our study provides insights into ChatGPT's performance and constraints when evaluated against the AI-powered EdTech code generation and auto-grading tools used in programming higher education for some time (i.e., *Kattis*). The results show that ChatGPT performs well in solving only easy-level programming tasks, which indicates its limited capability to be used by students to pass introductory programming courses. The findings contribute to the ongoing debate on the utility of AI-powered tools in introductory programming education.

ACKNOWLEDGMENTS

This research has been funded by Digital Futures, Stockholm, Sweden, and is part of the project "Responsible Digital Assessment Futures in Higher Education".

REFERENCES

[1] Yazid AlBadarin, Markku Tukiainen, Mohammed Saqr, and Nicholas Pope. 2023. A Systematic Literature Review of Empirical Research on ChatGPT in Education. *Available at SSRN 4562771* (2023).

[2] Ram B Basnet, Tenzin Doleck, David John Lemay, and Paul Bazelais. 2018. Exploring computer science students' continuance intentions to use Kattis. *Education and Information Technologies* 23 (2018), 1145–1158.

[3] Brett A Becker, Paul Denny, James Finnie-Ansley, Andrew Luxton-Reilly, James Prather, and Eddie Antonio Santos. 2023. Programming is hard-or at least it used to be: Educational opportunities and challenges of ai code generation. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*. 500–506.

[4] Cecilia Ka Yuk Chan and Wenjie Hu. 2023. Students' Voices on Generative AI: Perceptions, Benefits, and Challenges in Higher Education. *arXiv preprint arXiv:2305.00290* (2023).

[5] Hayden Cheers, Yuqing Lin, and Shamus P Smith. 2021. Academic source code plagiarism detection by measuring program behavioral similarity. *IEEE Access* 9 (2021), 50391–50412.

[6] Tyne Crow, Andrew Luxton-Reilly, and Burkhard Wuensche. 2018. Intelligent tutoring systems for programming education: a systematic review. In *Proceedings of the 20th Australasian Computing Education Conference*. 53–62.

[7] Yun Dai, Ang Liu, and Cher Ping Lim. 2023. Reconceptualizing ChatGPT and generative AI as a student-driven innovation in higher education. (2023).

[8] Emma Enström, Gunnar Kreitz, Fredrik Niemelä, and Viggo Kann. 2010. Testdriven utbildning–strukturerad formativ examination. In *NU2010, Dialog för lärande. Stockholm, Swe. 13-15 oktober 2010.*

[9] Borja Fernandez-Gauna, Naiara Rojo, and Manuel Graña. 2023. Automatic feedback and assessment of team-coding assignments in a DevOps context. *International Journal of Educational Technology in Higher Education* 20, 1 (2023), 17.

[10] Tim Fütterer, Christian Fischer, Anastasiia Alekseeva, Xiaobin Chen, Tamara Tate, Mark Warschauer, and Peter Gerjets. 2023. ChatGPT in Education: Global Reactions to AI Innovations. (2023).

[11] Dragan Gašević, George Siemens, and Shazia Sadiq. 2023. Empowering learners for the age of artificial intelligence. *Computers and Education: Artificial Intelligence* (2023), 100130.

[12] Chuqin Geng, Zhang Yihan, Brigitte Pientka, and Xujie Si. 2023. Can ChatGPT Pass An Introductory Level Functional Language Programming Course? *arXiv preprint arXiv:2305.02230* (2023).

[13] Reza Hadi Mogavi, Chao Deng, Justin Juho Kim, Pengyuan Zhou, Young D Kwon, Ahmed Hosny Saleh Metwally, Ahmed Tlili, Simone Bassanelli, Antonio Bucchiarone, Sujit Gujar, et al. 2023. Exploring User Perspectives on ChatGPT: Applications, Perceptions, and Implications for AI-Integrated Education. *arXiv e-prints* (2023), arXiv–2305.

[14] Lingchen Huang, Huazi Zhang, Rong Li, Yiqun Ge, and Jun Wang. 2019. AI coding: Learning to construct error correction codes. *IEEE Transactions on Communications* 68, 1 (2019), 26–39.

[15] Nayab Iqbal, Hassaan Ahmed, and Kaukab Abid Azhar. 2022. Exploring teachers' attitudes towards using chatgpt. *Glob. J. Manag. Adm. Sci* 3 (2022), 97–111.

[16] Sung-Hee Jin, Kowoon Im, Mina Yoo, Ido Roll, and Kyoungwon Seo. 2023. Supporting students' self-regulated learning in online learning using artificial intelligence applications. *International Journal of Educational Technology in Higher Education* 20, 1 (2023), 1–21.

[17] Majeed Kazemitabaar, Justin Chow, Carl Ka To Ma, Barbara J Ericson, David Weintrop, and Tovi Grossman. 2023. Studying the effect of AI Code Generators on Supporting Novice Learners in Introductory Programming. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–23.

[18] Natalie Kiesler and Daniel Schiffner. 2023. Large Language Models in Introductory Programming Education: ChatGPT's Performance and Implications for Assessments. *arXiv preprint arXiv:2308.08572* (2023).

[19] Weng Marc Lim, Asanka Gunasekara, Jessica Leigh Pallant, Jason Ian Pallant, and Ekaterina Pechenkina. 2023. Generative AI and the future of education: Ragnarök or reformation? A paradoxical perspective from management educators. *The International Journal of Management Education* 21, 2 (2023), 100790.

[20] Shuai Lu, Daya Guo, Shuo Ren, Junjie Huang, Alexey Svyatkovskiy, Ambrosio Blanco, Colin Clement, Dawn Drain, Daxin Jiang, Duyu Tang, et al. 2021. Codexglue: A machine learning benchmark dataset for code understanding and generation. *arXiv preprint arXiv:2102.04664* (2021).

[21] Iain Milne and Glenn Rowe. 2002. Difficulties in learning and teaching programming—views of students and tutors. *Education and Information technologies* 7 (2002), 55–66.

[22] Marta Montenegro-Rueda, José Fernández-Cerero, José María Fernández-Batanero, and Eloy López-Meneses. 2023. Impact of the Implementation of ChatGPT in Education: A Systematic Review. *Computers* 12, 8 (2023), 153.

[23] Davy Tsz Kit Ng, Jac Ka Lok Leung, Jiahong Su, Ross Chi Wui Ng, and Samuel Kai Wah Chu. 2023. Teachers' AI digital competencies and twenty-first century skills in the post-pandemic world. *Educational technology research and development* 71, 1 (2023), 137–161.

[24] Radek Pelánek. 2016. Applications of the Elo rating system in adaptive educational systems. *Computers & Education* 98 (2016), 169–179.

[25] Jaromir Savelka, Arav Agarwal, Christopher Bogart, Yifan Song, and Majd Sakr. 2023. Can Generative Pre-trained Transformers (GPT) Pass Assessments in Higher Education Programming Courses? *arXiv preprint arXiv:2303.09325* (2023).

[26] Priynka Sharma and Mayuri Harkishan. 2022. Designing an intelligent tutoring system for computer programing in the Pacific. *Education and Information Technologies* 27, 5 (2022), 6197–6209.

[27] Jennifer L Steele. 2023. To GPT or not GPT? Empowering our students to learn with AI. *Computers and Education: Artificial Intelligence* 5 (2023), 100160.

[28] Miriam Sullivan, Andrew Kelly, and Paul McLaughlan. 2023. ChatGPT in higher education: Considerations for academic integrity and student learning. (2023).

[29] Haoye Tian, Weiqi Lu, Tsz On Li, Xunzhu Tang, Shing-Chi Cheung, Jacques Klein, and Tegawendé F Bissyandé. 2023. Is ChatGPT the Ultimate Programming Assistant–How far is it? *arXiv preprint arXiv:2304.11938* (2023).

[30] Dwayne Towell and Brent Reeves. 2010. From Walls to Steps: Using online automatic homework checking tools to improve learning in introductory programming courses. (2010).

[31] Zhihong Xu, Yingying Zhao, Jeffrey Liew, Xuan Zhou, and Ashlynn Kogut. 2023. Synthesizing research evidence on self-regulated learning and academic achievement in online and blended learning environments: A scoping review. *Educational Research Review* (2023), 100510.

[32] Ramazan Yilmaz and Fatma Gizem Karaoglan Yilmaz. 2023. The effect of generative artificial intelligence (AI)-based tool use on students' computational thinking skills, programming self-efficacy and motivation. *Computers and Education: Artificial Intelligence* (2023), 100147.

[33] Cynthia Zastudil, Magdalena Rogalska, Christine Kapp, Jennifer Vaughn, and Stephen MacNeil. 2023. Generative AI in Computing Education: Perspectives of Students and Instructors. *arXiv preprint arXiv:2308.04309* (2023).