Toward evaluating the impact of ChatGPT on learning data structures and algorithms: an experiment with undergraduate students.

Waldemar Ferreira
Unidade Acadêmica de Belo Jardim, UFRPE
Belo Jardim, Brazil
waldemar.ferreira@ufrpe.br

Sergio Soares
Center of Informatics, UFPE
Recife, Brazil
scbs@cin.ufpe.br

Abstract

A fundamental skill in software design is the ability to identify and select the most appropriate data structures for a given problem. This study investigates the impact of using ChatGPT for undergraduate students engaged in a data structures and algorithms (DSA) course during practical activities. Due to limited resources, this study covers only two topics of DSA: Binary Search Trees and AVL Trees. An experiment was conducted involving two groups of students: the ChatGPT group, who could use ChatGPT for all lab session assignments, and the experimental group, who used ChatGPT only on the first activities and relied solely on instructor-provided materials for the remaining assignments. The impact of ChatGPT usage was measured using questionnaires, final exam grades, and code similarity analysis. The results indicate that there are no significant differences in performance between students using ChatGPT and those relying exclusively on traditional materials. However, since we cover only two topics of DSA, more research is necessary.

Keywords

ChatGPT, Algorithms and Data Structure, Programming, Experiment

1 Introduction

Software Engineering (SE) education is inherently complex, as it encompasses a wide range of knowledge areas. According to the Software Engineering Curriculum Guidelines (SE2014) [17], there are at least ten major knowledge areas, including Computing Essentials, Software Design, Software Verification and Validation, among others. Still, according to SE2014, within the Computing Essentials area, a key component considered crucial is the study of algorithms, data structures, and computational complexity. This area focuses on the design, analysis, and implementation of efficient computational solutions, emphasizing how algorithmic strategies, data organization, and computational complexity affect software performance and scalability [5]. These competencies shape a professional's capacity for informed decisions about performance, scalability, and resource management. So that a professional can make informed decisions about algorithm and data structure selection and evaluate trade-offs among performance, scalability, and resource consumption in real-world applications.

In this context, generative Artificial Intelligence (AI) in education, particularly at the university level, has grown recently with a significant impact [19]. There are many implementations of generative AI, among them ChatGPT [3]. Tools, like ChatGPT, can aid students with homework, generate practice problems, and provide tutoring, making learning more affordable and personalized [25]. Some researchers have observed that the use of ChatGPT can: (i) boost students' interest and motivation [21], (ii) help computer science students by providing immediate answers to programming questions [24], and (iii) provide examples to explain complex programming concepts [25]. On the other hand, many fear widespread cheating that is hard, or even impossible, to catch [29], as well as concerns about the use of generative AI may imply on superficial learning [8].

A key limitation of ChatGPT is its reliance on the knowledge it was trained on [25], which can lead to inaccurate responses to complex questions [7]. Effective data structure analysis requires a deep understanding, which educators can offer through step-by-step explanations. However, maybe the most important drawback is that it may discourage students from developing critical skills such as reasoning [23]. Students who excessively rely on ChatGPT to generate code may not cultivate the ability to solve problems independently [23]. Consequently, students might fail to develop critical skills such as critical thinking, creativity, and decision-making [10]. These are crucial capabilities for SE, especially when involving DSA [34]. Excessive use and cheating are major concerns for SE educators. Some universities have even blocked access to the ChatGPT website on school grounds to mitigate these risks [37].

ChatGPT is here to stay, so we believe that, rather than avoiding them, we should embrace them to modernize education [37]. Some studies propose methods in which ChatGPT can serve as a Teaching Assistant (TA) [13] or generate class summaries [14]. However, in our experience, students often use it as an Oracle for coding exercises, suggesting a lack of problem-solving skills [10]. In this paper, we report an experiment evaluating the impact of using ChatGPT during coding tasks in a DSA course. This course is part of the third semester of the Computer Engineering undergraduate program at the Federal Rural University of Pernambuco, Belo Jardim-PE. Our experiment aims to investigate the impact of ChatGPT on: (i) learning outcomes via questionnaires about Data Structures and Algorithms, (ii) final exam performance, and (iii) code similarity in student lab submissions. Due to the limited time frame to plan, execute, and report results, we focused on only two topics from the DSA course: Binary Search Tree (BST) and AVL Tree [5]. In summary, we observed no statistically significant differences in questionnaire responses or final exam results. However, we found that the code submitted by the students who used ChatGPT was more likely to have high similarity to others. However, our results address only a limited subset of DSA topics; therefore, further research is needed to broaden the scope of investigation.

This paper is organized as follows. Section 2 provides a review of related work, highlighting previous studies on the usage of ChatGPT in computer science education, particularly in Data Structures and Algorithms' contexts. Section 3 details the methodology employed in this study, including the experimental design, participant selection, data collection procedures, and statistical methods used to evaluate the research hypotheses. Section 4 presents the results of the experiment, accompanied by a comprehensive analysis and interpretation of the data, with comparisons to existing literature. Section 5 discusses potential threats to the study's validity, including internal, external, construct, and statistical validity. Finally, Section 6 concludes the paper by summarizing the main findings, reflecting on their implications for education and practice, and outlining directions for future research.

2 Related Work

The recent popularity of generative AI has highlighted its potential benefits, such as AI-assisted pair programming [25], as well as its drawbacks, such as its facilitation of cheating [9]. These factors have drawn the attention of researchers worldwide to investigate the educational implications of generative artificial intelligence, including ChatGPT.

Garcia [11] conducted a rapid literature review synthesizing 107 studies on the use of ChatGPT in computer programming education. The review identified major application areas, including personalized tutoring, knowledge reinforcement, instructional material generation, source code production, immediate feedback, and assessment support. Findings indicate that ChatGPT can enhance learning efficiency and instructional support, particularly through individualized guidance and automated feedback. However, the review also highlights substantial challenges, such as academic dishonesty, ethical concerns, overreliance on AI tools, diminished critical thinking, and technical limitations. Garcia further noted that most existing studies emphasize basic programming instruction, while advanced topics such as DSA remain underexplored.

Pinto et al. [22] have thoroughly examined the impact of generative AI, such as ChatGPT, on education, highlighting its benefits and challenges. These studies show that AI-driven technologies can enhance student engagement and learning outcomes by providing personalized learning experiences, instant feedback, and detailed explanations. However, concerns about over-reliance on these tools, which may potentially undermine students' creativity and problem-solving abilities, persist. Additionally, ethical issues, such as AI bias and academic integrity, must be addressed. Pinto et al. observed mixed results: improved performance and motivation; however, minimal impact on learning outcomes. This research emphasizes the need for carefully designed educational frameworks and assignments to maximize the benefits of AI tools while mitigating potential drawbacks. They argue that further research is needed to explore these dynamics in various educational contexts and academic levels (such as DSA).

Some studies are conducted on complex topics beyond basic programming concepts. For instance, Abdulhadi [28] conducted an experiment in an Embedded Systems course. Their primary objective was to determine how effectively ChatGPT could assist students in answering quiz questions without prior learning of the related topics. Their findings indicated that the ChatGPT group performed better on code analysis and theoretical questions but struggled with code completion and image-based questions. However, their results for writing complex code were inconsistent. The author concluded that ChatGPT is currently insufficient for computer engineering programs.

Kosar et al. [16] investigate the impact of ChatGPT on the learning outcomes of undergraduate students in an object-oriented programming course. The study involved 182 participants divided into two groups: one using ChatGPT and the other not. They analyzed practical assignments, midterm exams, and final grades. Their results indicated that ChatGPT usage did not significantly affect students' performance. The feedback revealed that students found ChatGPT useful for code optimization and understanding, but did not rely on it excessively. The findings suggest that with appropriate measures, ChatGPT can be safely integrated into programming education without compromising learning outcomes.

Yilmaz et al. [36] investigate integrating ChatGPT into programming education, highlighting its potential benefits and limitations. Similar to the previous study, this study was also conducted in an object-oriented programming course. Their study found that ChatGPT can provide instant feedback, improve debugging, and enhance students' confidence and understanding of complex topics. However, challenges such as fostering dependency, potential inaccuracies, and ethical concerns were also noted. The research suggests that while generative AI tools like ChatGPT can significantly aid programming education, educators should be aware of these tools' limitations and take appropriate measures to ensure they complement, rather than hinder, the educational process. Additionally, they said that further studies are needed to explore the varied impacts of these tools across different programming courses and educational levels. They also reported that some participants noted drawbacks, including potential laziness, weakened cognitive skills, and the risk of incorrect answers. Another related study by [24] used a between-subjects design with two groups: one using ChatGPT and the other relying on textbooks and notes without internet access. Unlike our study, which included assignments throughout the course, their participants faced programming challenges after the course. They found that the ChatGPT group achieved higher scores faster on defined tasks but struggled with more complex problems, showing inaccuracies and inconsistencies.

To the best of our knowledge, Jaime et al. [13] is the only study that specifically investigates the impact of ChatGPT on DSA. This work investigates integrating ChatGPT as a supplementary tool for teaching assistants (TAs). It aims to enhance teaching effectiveness and student learning outcomes. Through a controlled experiment involving 40 undergraduate students divided into a traditional TA group and a ChatGPT-assisted group. The researchers found that using ChatGPT — guided by structured prompts and verified by TAs — led to significantly better performance. The study emphasizes the benefits of combining human oversight with AI-generated content, noting improvements in engagement, problem-solving, and understanding of complex topics like recursion and dynamic programming. This study motivated our work by highlighting the impact of generative AI in DSA. Mastery of this subject typically requires extensive practice through problem-solving. However, students often appear to use ChatGPT not as a TA, but rather as an "oracle" that directly provides solutions to exercises. It is this pattern of use that the present study seeks to investigate. We encouraged students to use ChatGPT freely in their usual manner, while Jaime et al. implemented a more constrained approach in which students interacted with ChatGPT solely through TA-curated, structured prompts..

3 Method

This study evaluates how ChatGPT influences undergraduate students' learning of DSA. We specifically focus on the effects of unrestricted use of ChatGPT, allowing students to use the tool self-directedly. To direct our inquiry, we test four hypotheses regarding ChatGPT's adoption, its effect on comprehension, final grades, and code similarity among students.

- $H0_0$: There is no significant difference in the use of ChatGPT among students when responding to DSA exercises.
- $H0_1$: There is no significant difference in students' comprehension when using ChatGPT vs. when not.
- $H0_2$: There is no significant difference in students' final lab grades between those who used ChatGPT and those who did not.
- $H0_3$: There is no significant difference in students' code similarity between those using ChatGPT and those who do not.

As noted, only the first hypothesis, $H0_0$, cannot be evaluated using a statistical model; therefore, it is examined solely through students' survey responses (see Section 3.2 for details). To evaluate comprehension of hypothesis $H0_1$, we analyzed students' responses from the pretest and posttest questionnaires (see Section 3.2 for details). Hypothesis $H0_2$ was assessed using the final course grades of the course. For the analysis of hypothesis $H0_3$, we employed the MOSS (Measure of Software Similarity) [1] tool to examine code similarity. MOSS is a widely recognized system for detecting program similarities, effectively identifying instances of code reuse and potential plagiarism [4]. By applying MOSS to the students' source code, we systematically assessed the degree of similarity across different submissions, enabling us to rigorously evaluate the originality and independence of each student's work. We utilized MOSS to analyze the artifacts from both groups.

3.1 The Data Structures and Algorithms Course

As previously noted, Data Structures and Algorithms (DSA) are among the fundamental topics identified in the Software Engineering Curriculum [17]. These guidelines classify *algorithms, data structures, and complexity* under the category of Computing Essentials (CMP.cf.2). Besides, in accordance with Brazilian educational legislation, DSA constitute a mandatory component of all undergraduate engineering computing programs [2].

This study was conducted with students enrolled in a Bachelor of Computer Engineering program at a Brazilian university. Within our curriculum, the course is offered during the third semester and requires Introduction to Programming and Object-Oriented Programming as prerequisites. Both subjects are also mandated by Brazilian educational regulations[3].

The course adopted Java as the programming language, with students having been previously exposed to it in Object-Oriented Programming courses. The decision to use Java over other programming languages was made by the faculty council. The limitations associated with the use of Java in this study are discussed in Section 5.

The content of DSA courses varies widely across undergraduate programs worldwide [32]. Owing to constraints on complexity and available resources, this study did not cover all topics in DSA. In fact, it did not cover the entire course syllabus of our program. Instead, the investigation focused on two specific topics: (i) Binary Search Trees and (ii) AVL trees. These topics were selected for the following reasons:

(1) They are common topics across many undergraduate curricula. For example, both are included in the Bachelor of Science in Software Engineering program at the University of Mississippi, as reported in Appendix A of the Software Engineering Curriculum Guidelines (SE2014) [17].
(2) These topics are well-suited to the constraints of our experimental design (see Section 3.2), making them representative and pedagogically relevant for evaluating ChatGPT's impact in this educational context.
(3) They constitute two core chapters in a widely used data structures and algorithms textbook, the Introduction to Algorithms by Cormen et al. (Chapters 11 and 12) [5].

A Binary Search Tree (BST) is a data structure used to store elements in a hierarchical order that allows for efficient searching, insertion, and deletion operations [5]. In a BST, each node contains a key, and it satisfies the binary search property: the key in the left child is less than the key in the parent node, and the key in the right child is greater. This recursive structure enables high-efficiency search and insertion when balanced.

An AVL tree is a self-balancing variant of the BST, designed to maintain a balanced structure and thus ensure consistently efficient operations [5]. The AVL tree maintains the property that the height difference (balance factor) between the left and right subtrees of any node is at most one. After every insertion or deletion, the tree performs specific rotations (single or double) to restore balance if needed. This guarantees that the tree's height remains logarithmic in the number of nodes, ensuring great performance for search, insertion, and deletion in all cases.

The DSA course was held at the end of 2024 and early 2025, during which **GPT-4o Mini** was the only ChatGPT version available to students.

3.2 Experiment Phases

This research investigates how ChatGPT impacts undergraduate learning in data structures and algorithms. We conducted a pretest-posttest control group experiment, following two established guidelines [15, 35]. Forty-three students of a Brazilian university participated in the study. As mentioned earlier, this study followed a pretest-posttest design across four phases, as shown in Figure 1.

[1]https://theory.stanford.edu/~aiken/moss/
[2]Resolução CNE/CES nº 5, de 16 de novembro de 2016
[3]Resolução CNE/CES nº 5, de 16 de novembro de 2016

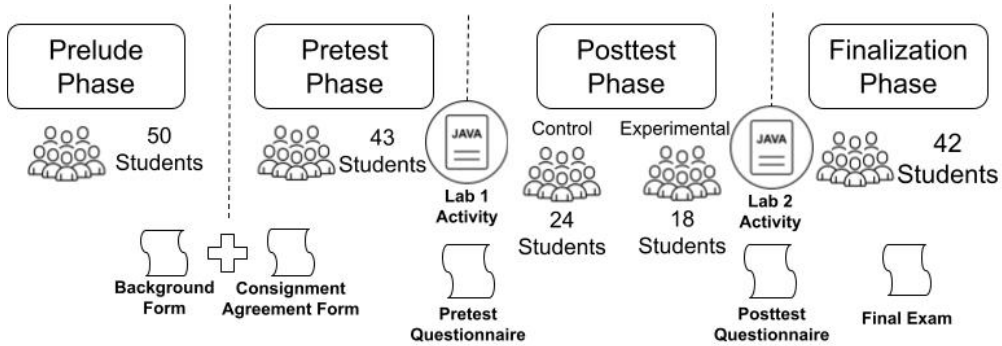

Figure 1: Experiment Process Overview.

In the first phase (Prelude Phase), the experiment began with an introductory session where the lecturer (the first author) outlined the study and its requirements. As required by the ethical standards [12], participation was not mandatory. To incentivize students, the lecturer offered an extra point in the final exam to those who participated in the entire process. Those students who agreed to participate had to sign an informed consent form, ensuring compliance with ethical research standards [33].

A total of 50 students agreed to participate in the experiment and signed the informed consent form. After this, the study progressed to the Pretest Phase (Figure 1), which spanned two class sessions: the first theoretical and the next practical. In the theoretical class, the lecturer spent two hours teaching Binary Search Tree (BST), focusing solely on conceptual content. During this phase, participants were not yet divided into control and experimental groups.

In the subsequent class, students were instructed to implement the BST concepts discussed in the previous lecture using Java. They were given two hours to complete this programming task. The lecturer developed an environment to validate students' implementations. This environment corresponds to a GitHub repository with skeleton code (stubs) and unit tests. All students were encouraged to fork the repository and implement their solutions. The main fork served as a guideline for constructing practical assignments. The code provided to the students is available in the Artifact Availability section.

It is important to note that during the Pretest Phase, all students were allowed to use ChatGPT during the lab session. The lab assignments were designed solely based on the textbook [5], but the lecturer observed that the students were able to solve them after a few prompts.

After finishing the lab assignments by submitting their implementations, students were asked to complete a questionnaire (the Pretest Questionnaire in Figure 1). This questionnaire aims to assess their understanding of the BST topic. It contains questions about both theoretical and practical aspects of BST. We gave 30 minutes to answer the questionnaire. Only students who completed the questionnaire were allowed to proceed to the next phase of the experiment.

A few students encountered difficulties completing the lab activity within the allotted time (2 hours). For those who did not succeed, the instructor permitted them to submit their implementation after class. However, the questionnaire had to be completed during the final 30 minutes of the class. Students were not permitted to use ChatGPT while completing the questionnaire.

The pretest questionnaire consisted of 22 multiple-choice questions. All participants finished within the allocated time, and no complaints about time constraints were reported. Of the 50 students who consented to participate in the experiment, 49 attended both the theoretical and practical classes, submitted their implementations, and completed the questionnaire. Only one participant did not attend one class, so we excluded these data and considered him as a dropout [35].

After the pretest questionnaire was administered, the Posttest Phase began (Figure 1). This phase covered AVL trees. This phase followed the same procedure as the pretest phase. Initially, the teacher conducted a lecture covering the fundamental concepts of AVL trees. In the following class, the practical task began, during which students were granted access to a GitHub repository containing the lab assignment. As in the previous phase, all students were encouraged to fork the repository and implement an AVL tree using the concepts introduced in the theoretical class.

Before the practical activity was available, participants were randomly assigned to two groups. To ensure group equivalence, we analyzed the pretest questionnaire results to determine whether the groups had similar responses. Further details of this analysis are provided in Section 4.2. The groups were:

- The Control Group consisted of students who, as in the previous lab session, were allowed to use ChatGPT throughout the lab activity without any restrictions. They could access ChatGPT for help, guidance, or code suggestions while completing the assignment.
- The Experimental Group comprised students who were not allowed to use ChatGPT at the lab activity (AVL tree). Instead, these students were encouraged to consult only lecture notes, official documentation of relevant application libraries, and example materials. They were also allowed to use internet search engines like Google, but not any AI-enabled response platforms.

As in the previous phase, the lab assignment lasted two hours. Students were allowed to complete the posttest questionnaire only after submitting their lab assignments. The questionnaire consisted of 21 multiple-choice questions, with a maximum completion time of 30 minutes. As in the last phase, participants in both groups were prohibited from using ChatGPT during questionnaire completion. As with the earlier questionnaire, all students completed it within the allotted time, and no complaints were reported. Unlike in the pretest phase, all students completed the lab assignment within the designated two-hour session.

To distinguish students by group, the lecturer brought two sets of Post-it notes, green and yellow, and placed a note on each student's monitor to indicate group assignment. The lecturer continuously monitored those students who were not permitted to use ChatGPT. Any student attempting to access the tool was immediately instructed not to use it and to rely exclusively on the materials provided by the lecturer.

During the lab session, six students were unable to complete the lab assignment and had to leave before the session ended. As a result, they were not permitted to complete the posttest questionnaire. We classified them as dropouts. Of these students, one belonged to the Control Group and seven to the Experimental Group. Their responses to the pretest questionnaire were also excluded from the analysis. The remaining participants were distributed as follows: 24 students in the Control Group and 18 students in the Experimental Group.

4 Result and Discussion

This section presents the results derived from the collected data. A total of 42 students participated in the entire experiment, yielding 42 observations in all experiment phases. This sample size exceeds the commonly recommended minimum threshold of 30, which is generally considered sufficient to improve statistical power and reduce the risk of Type II errors [27]. Further discussion of sample size limitations is provided in Section 5. Table 1 presents a summary of the collected data.

The right-hand side of Table 1 presents demographic information from our sample. We included only data from students who participated in the two phases of our experiment (Full Participation). As we can see, the majority of participants are regular students enrolled in Data Structures and Algorithms in the third semester, as recommended by our program. The average age of participants was approximately 22 years. Regarding gender identity, 83% identified as men and 17% as women; no participants selected other gender identities or chose to withhold this information. Given its characteristics, we consider that our sample is representative of the typical undergraduate student in a computer engineering program [18].

All observations were statistically tested with an alpha level of 0.05 as the significance threshold [27]. Given the small sample size, we did not apply the Shapiro–Wilk test, because we did not presume normality [15]. We used the Mann-Whitney U test [15], a nonparametric test for two independent samples (because the data are nonnormal). To assess the effect size, we used Cliff's Delta [15].

4.1 Familiarity of Students with ChatGPT in Coding activities

Our first inquiry was to investigate whether the students know generative AI, especially ChatGPT. Moreover, if and how they use it in programming activities. All participants reported familiarity with generative AI and ChatGPT.

Apart from whether they know generative AI and ChatGPT, they were also asked how frequently they use it in coding and in everyday activities. Figure 2 presents our results. As shown in Figure 2.a, the majority of participants (81%) reported often/always usage of generative AI for programming tasks. However, only four students said they always use it. The other four participants stated they rarely use generative AI for such tasks, and notably, none reported using it.

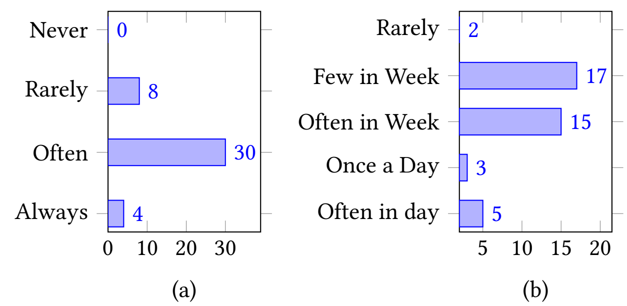

Figure 2: An overview of how often students use generative AI in coding (a) and in everyday activities (b).

Considering Figure 2.b, we conclude that the use of generative AI is not a panacea. Students do not appear to use them extensively in everyday activities, unlike their frequent use of such tools for coding-related tasks (Figure 2.a). It is important to note that this experiment was conducted at the end of 2024 and the beginning of 2025, a period during which AI agents had not yet been widely integrated into common platforms, such as Gemini in Google Search.

Finally, participants were asked which generative AI tools they were familiar with. All respondents reported knowing ChatGPT; 32 indicated awareness of Microsoft Copilot; 19 mentioned Google Bard (now Google Gemini); and 2 cited Bing AI. The high prevalence of ChatGPT may be attributed to the recent release and widespread visibility of GPT-4.0o at the time of the study, as well as the subsequent migration of Bing AI to GPT-based models [1]. In addition, the relatively high awareness of Microsoft Copilot may stem from two factors: (i) it was specifically developed to support code development, and (ii) it is integrated into Visual Studio Code, one of the most widely used development environments. In addition, several prominent generative AI tools, such as Claude and DeepSeek, were not yet widely available to the public during the study period [1].

Our results indicate that ChatGPT is both highly familiar and frequently used among participants, particularly for programming tasks, with nearly 80% reporting consistent use. This suggests that ChatGPT has become a central tool in students' academic workflows, especially in computing-related courses. This finding aligns with the observations of Daun et al. [6], who report that ChatGPT is increasingly regarded as a panacea in software engineering. Our findings are consistent with the Stack Overflow survey [31], both in terms of the frequency of generative AI use and the most commonly adopted tools.

Table 1: Data summary.

Students					Full Participation Data			
Enrolled in Course	Agreed in participate	Dropout		Full Participation	Semester	Age	Gender	
		Pretest Phase	Posttest Phase				Male	Female
57	50	1	7	42	3.37	22.21	25 (83%)	5 (17%)

4.2 Ensuring Group Equivalence

Before testing any hypotheses, it is essential to verify that the participant groups are balanced. As described in Section 3, participants were randomly assigned to two groups. However, some students were unable to complete the lab assignments (Table 1, column Dropout). This imbalance could potentially introduce bias into the results. To assess whether it affected group comparability, we conducted a statistical analysis to determine whether there were significant differences in pretest scores between the two groups, excluding the dropout data. The results of this analysis are presented in Table 2.

As shown in Table 2, the results indicate that neither the p-value (1.0) nor Cliff's Delta (0.0) suggests a statistically significant difference between the groups. Remember that, in the pretest phase, all students performed the same lab assignment, answered the same questionnaire, and used ChatGPT. This result suggests that participants in both groups exhibited a comparable understanding of the theoretical and practical aspects of BST. Therefore, despite the unequal group sizes, the balance between groups appears to have been maintained and is unlikely to have biased the results.

4.3 Posttest ChatGPT Group vs. Non ChatGPT Group

To test hypothesis $H0_1$, we compared posttest questionnaire responses from participants who used ChatGPT with those from participants who did not. The results of this comparison are presented in Table 3.

Analyzing the data presented in Table 3, we find no sufficient evidence to reject hypothesis $H0_1$. Therefore, we conclude that ChatGPT usage does not have a statistically significant impact on students' comprehension. At first glance, this result appears to contrast with the findings of Daun et al. [6], who reported higher performance among students who used ChatGPT as a teaching assistant. However, their analysis was based solely on average scores across various assessments (e.g., exercises, quizzes, midterms, and final exams). In other words, the authors did not employ a statistical model to support their arguments. If a similar approach is applied in our context (considering only the mean post-test questionnaire scores), we obtain a similar result. The result is that the ChatGPT group achieved an average score of 4.94, slightly higher than the 4.89 of the non-ChatGPT group. It could mean that they answered two more questions correctly than the other group. Nevertheless, this difference is minimal and not statistically significant.

One potential factor that could have biased the previously discussed result is the higher dropout rate observed in the Experimental Group (i.e., the students who did not use ChatGPT during the posttest phase). As discussed in Section 4.2, our analysis indicated that the groups remained balanced despite these dropouts. However, to further ensure that this did not affect our findings, we conducted an additional analysis comparing the Experimental Group's performance in the pretest and posttest phases. This comparison aimed to determine whether the same group of students performed differently when ChatGPT was or was not available. The results of this analysis are presented in Table 4.

It is important to note that, unlike the other hypotheses for which we applied the Mann–Whitney U test, the Wilcoxon signed-rank test was used to evaluate this hypothesis [15]. This choice was based on the nature of the data, as the Wilcoxon test is more appropriate for paired samples. In this case, we compared pretest and posttest questionnaire responses from the same group of participants.

As shown in Table 4, there is no statistically significant difference in questionnaire answers when comparing the same group of students when they can or cannot access ChatGPT during lab sessions (p-value equals to 0.86504). However, a slight decrease in the mean score was observed when participants did not use ChatGPT, suggesting they answered fewer questions correctly under this condition. The difference in mean scores is approximately 0.1 point, indicating that students answered about two more questions correctly when using ChatGPT during lab sessions. This finding is consistent with Daun et al. [6], who reported improved performance among students who used ChatGPT. Nevertheless, as in our previous analysis, the observed difference in means is not statistically significant. This may indicate the need for further testing with a larger sample size, or that ChatGPT's presence has a limited (or none) impact in this context.

Another factor to consider is the potential influence of the instructor or lecture material, which may have obscured the effect of ChatGPT usage. To investigate this possibility, we analyzed whether there was a significant difference between pretest and posttest scores within the Control Group (which used ChatGPT in both phases). The results of this analysis are presented in Table 5.

These results show that there is no statistically significant difference in scores when students are allowed to use ChatGPT to support their lab activities, regardless of whether the topic was BST or AVL trees. Unsurprisingly, this comparison yielded the smallest difference in mean scores (only 0.076), suggesting that both the lecture materials and the instructor's influence were consistent when teaching BST.

An important observation from tables 3, 4, and 5 is that all reported Cliff's Delta values are close to zero (respectively, -0.05787, -0.08333, and -0.15278). These values indicate negligible effect sizes, suggesting that the differences between the two groups are minimal. This result may reflect two possibilities: (i) the effect of ChatGPT usage is limited, or (ii) the sample size is insufficient to reveal a measurable effect, with the evidence more strongly supporting the latter explanation.

Table 2: Pretest results between participants in each group.

Groups	Participants	Mean	Variance	U Statistic	p-value	Cliff's-delta
Control Group	24	4.864	.76	216	1.0	0.0
Experimental Group	18	5.035	1.942	-	-	-

Table 3: Posttest ChatGPT Group vs. Non ChatGPT Group.

Groups	Participants	Mean	Variance	U Statistics	p-value	Cliff's-delta
Control Group	24	4.94	1.335	203	0.77249	-0.05787
Experimental Group	18	4.894	1.782	-	-	-

Table 4: Comparison of pretest and posttest results of Experimental Group (Wilcoxon signed-rank).

Groups	Participants	Mean	Variance	Test Result	p-value	Cliff-delta
Experimental Group Pretest	18	5.035	1.942	81	0.86504	-0.08333
Experimental Group Posttest	18	4.894	1.782	-	-	-

Table 5: Comparison between pretest and posttest results of the Control Group.

Groups	Participants	Mean	Variance	U Statistics	p-value	Cliff's delta
Control Group pretest	24	4.864	0.76	144	0.87741	-0.15278
Control Group posttest	24	4.94	1.335	-	-	-

4.4 Comparing Final Exam Scores

Hypothesis $H0_2$ examines the potential impact of ChatGPT usage on students' knowledge and grades. Table 6 displays the statistical analysis of students' scores on the final exam. This exam covered topics beyond BSTs and AVL Trees, namely sorting algorithms and hash tables [5]. It is important to clarify that the grades at our university are on a scale of 0.0 to 10.0.

Hypothesis $H0_2$ investigates the potential impact of ChatGPT usage on students' overall knowledge acquisition and academic performance. Table 6 presents statistics about the data collected from the students' final exam scores. This exam assessed a broader range of topics beyond BST and AVL Trees. It includes sorting algorithms and hash tables [5].

Based on the data presented in Table 6, we find no sufficient evidence to reject hypothesis $H0_2$. Therefore, we cannot conclude that using ChatGPT when implementing lab assignments has a significant effect on students' understanding of the specific topic content. Additionally, the high variance in exam scores limits the reliability of any definitive conclusions drawn from this analysis. Nonetheless, the result is consistent with our previous findings, further suggesting that ChatGPT usage does not substantially influence academic performance in this context.

A crucial observation is that the effect is very small, as indicated by the Cliff's Delta value of 0.03704. An important factor that may have influenced this result is that the final grade encompasses multiple topics beyond BST and AVL Trees. We acknowledge this as a potential source of bias. However, the final examination consists of an interconnected set of questions, making it impractical to isolate responses exclusively related to BSTs and AVL Trees, since these questions depend on solutions to earlier problems involving topics such as sorting algorithms.

4.5 Comparing Similarity Between Code Submitted by ChatGPT vs. Non-ChatGPT Groups

Our final hypothesis, $H0_3$, investigates whether the source code produced by students during lab sessions is more likely to be similar to others when using ChatGPT. Table 7 presents the summary statistics for the data collected from the MOSS analysis of the delivered students' source code.

Unlike previous findings, these results indicate that the code produced in lab sessions by the ChatGPT group is more similar to that produced without it. This conclusion is supported by a p-value less than 0.05. Besides, with a Cliff's delta of 0.43, we can conclude that the treatment had a **medium** influence. Thus, we can reject the null hypothesis $H0_3$, concluding that students who use ChatGPT tend to produce more similar source code than those who do not. This observed similarity is likely due to ChatGPT being trained on publicly available source code. Since BST and AVL tree problems are standard topics in computer science, standard implementations (often derived from textbooks) are widely accessible online. Consequently, ChatGPT may reproduce these common patterns. However, when applied to novel or less conventional problems, the training data may lack sufficient diversity or quality, potentially leading

Table 6: Statistics of Final Exam scores between the groups.

Groups	Participants	Mean	Variance	U Statistics	p-value	Cliff's delta
Control Group	24	7.202	4.86	224	0.85046	0.03704
Experimental Group	18	7.05	6.914	-	-	-

Table 7: Comparison of MOSS similarity between the groups.

Groups	Participants	Min	Max	Mean	Variance	U Statistics	p-value	Cliff's delta
Control Group	24	64.0%	100.0%	76.62%	103.8	309	0.01755	0.43056
Experimental Group	18	54.3%	89.9%	68.08%	89.76	-	-	-

the model to learn incomplete or incorrect patterns and generate inaccurate or suboptimal responses [9].

A notable finding is that students who did not use ChatGPT were less likely to produce code similar to that produced by students who did, suggesting they explored a wider range of approaches to solve the same problem. This may indicate a deeper understanding of the content, as these students were more likely to construct solutions independently rather than adapting available code. This observation aligns with Fuchs' argument [9], which highlights the potential risk of overreliance on generative technologies like ChatGPT, potentially hindering the development of critical thinking and other higher-order cognitive skills. On the other hand, it is important to note that our findings related to the other hypotheses do not provide any evidence to support this inference conclusively (Sections 4.4 and 4.3).

Finally, it is important to note a high similarity in the source code across both groups (Table 7). It is not a surprise since the instructor provides, before the lab session starts, a stub of the data structure (BST and AVL Tree) and a set of JUnit tests to verify the correctness of the developed source code. Students must write the empty methods in accordance with the theory.

4.6 Lesson Learned

Some universities have chosen to prohibit or restrict students' use of generative AI tools [37]. This situation echoes earlier concerns raised in the late twentieth century regarding calculators, which were initially resisted on the grounds that they would make students mathematically incompetent (a fear that proved unfounded). Generative AI tools are likely to remain a permanent presence in education, requiring instructors to adapt their pedagogical practices accordingly. Based on our experience conducting this study, we offer the following lessons learned:

(1) **Separate theoretical and practical assessments.** Since we administered only a single questionnaire for each topic, we were unable to analyze potential differences specifically in practical learning outcomes. Furthermore, all students attended the same theoretical lectures, regardless of whether they used ChatGPT during the laboratory sessions. This shared theoretical instruction may have obscured differences attributable to the intervention, potentially masking ChatGPT's impact on students' practical learning. And it may lead us not to reject some null hypotheses.

(2) **Avoid standard code-completion tasks.** Such exercises are well within the capabilities of current generative AI tools, which often produce highly accurate solutions. Instead, instructors should design problems: (i) with richer contextual requirements, multi-step reasoning, visual components [28], or (ii) content drawn from recent programming contests, which are less likely to be well represented in training data [38].

(3) **Avoid textbook-based exercises.** Solutions to many textbook problems are available online through forums or publishers' resources [5]. The generative AI has already incorporated them into its training data. This reduces their pedagogical value and may discourage genuine student engagement. Consequently, teachers need to reconceptualize these activities by embedding them in more complex, context-rich scenarios.

(4) **Try assignments using generative AI tools.** Before distributing tasks to students, instructors have to evaluate them using generative AI tools such as ChatGPT to understand how easily solutions can be generated and to refine prompts accordingly.

(5) **Integrate generative AI as a teaching assistant.** Rather than excluding these tools, educators should incorporate them into instruction by demonstrating effective prompt formulation and appropriate use. It is equally important to present counterexamples that illustrate the limitations and potential errors of generative AI systems, reinforcing the need for critical evaluation [2].

5 Threats to Validity

A typical threat to internal validity involves the potential for bias introduced by the way the experiment is conducted [35]. To mitigate this risk, we ensured consistency across both groups by using the same instructor, teaching materials, and laboratory assignments. This approach was intended to isolate the intervention as the only variable, minimizing the influence of any extraneous factors on the experimental outcomes.

Pretest and post-test designs are widely employed across domains such as education, healthcare, psychology, and marketing [15]. Pretest–posttest designs are often preferred over AB designs because they provide baseline measurements that enable the assessment of individual changes over time, offer greater control over potential confounding variables, and improve statistical power. In the context of this study, this design enabled us to evaluate both the equivalence of prior experience across groups and the consistency of the instructional materials, including the theoretical lectures, GitHub repository, and questionnaires.

Researchers may unintentionally introduce bias into statistical analyses due to preconceived expectations [35]. To address this threat, the second author anonymized the collected data before analysis, ensuring that the first author performed all statistical evaluations without knowing which data belonged to students who used ChatGPT and which did not. Accordingly, all data presented in the Artifact Availability Section refer only to Group A and Group B, without disclosing group identities. Moreover, the confidentiality of students' personal information and research data was rigorously preserved throughout the study.

Another common threat in extended experiments is participant fatigue [35]. To mitigate this risk, each theoretical lecture and laboratory session was limited to 2 hours. Additionally, these sessions were never held on the same day. They have a minimum interval of one day. Besides, students were instructed to complete the lab assignments within the allocated two-hour period. Only a small number of students were unable to finish within this timeframe and completed the assignments at home. While this may have introduced a minor source of bias, the number of such cases was limited and unlikely to have significantly impacted the overall findings.

The complexity of the questionnaires could pose another threat to validity. However, both questionnaires (pretest and posttest) were derived from the same data set [4]. Additionally, the order of questions was randomized for each participant during the administration of both questionnaires, reducing the likelihood of answer copying and preserving the integrity of individual responses.

The language adopted for the course, Java, poses a potential threat to validity. Results may differ if comparing Java to other languages due to the extensive availability of Java code. Java is one of the most widely used languages globally [20], resulting in a vast amount of code for BST and AVL Trees that can be used as input for ChatGPT. If we had used another language, such as Python, or a more recent one, such as TypeScript, the scarcity of materials on data structures in those languages might have influenced our results differently. However, the university's faculty council chose Java.

There is a potential risk that participants in the Control Group also used ChatGPT. To mitigate this threat, the first author monitored the implementation of lab session assignments. During the lab session, we labeled each student's computer to identify clearly which students could or could not use ChatGPT.

One potential threat to validity is the level of student motivation to fully engage with the experimental tasks [35]. To address this concern, an incentive of one additional point in the final grade was offered to students who participated in the study. However, this bonus was awarded only to those who attended all lectures and submitted the required lab assignments, thereby encouraging consistent and active participation throughout the experiment.

Additionally, the time allotted to complete the questionnaires was limited to 30 minutes. Some critics can argue that the time constraints can lead to rushed responses [35]. In our experience, 30 minutes appeared sufficient for answering 21–22 questions; all items were concise, multiple-choice questions that did not require extended reasoning or written explanations, thereby reducing cognitive load. Besides, the first author conducted pilot tests to evaluate timing with students who had already taken the course, and the duration was deemed appropriate. Furthermore, as noted in Section 3.2, no students reported any concerns regarding the allotted time.

Our sample size is relatively small [15], which limits the extent to which our findings can be generalized to a broader population. However, we applied statistical tests that are appropriate for small-sample studies [15] to ensure the validity of our analysis.

Finally, this experiment involved only second-year students, which poses a threat to external validity when generalizing to students from other academic years [35]. This focus was intentional, as our study specifically targeted novice programmers. Another threat to external validity concerns the applicability of our findings beyond the domain of software engineering. It remains unclear how the results would differ if the experiment were conducted in the context of other courses during the same academic period.

The validity of the measurements poses a potential threat to this study, as the questionnaires do not provide sufficient evidence of construct validity, item quality, or internal consistency, which may weaken conclusions about learning outcomes. Nevertheless, we attempted to mitigate this threat by adopting questionnaires derived from a well-established educational platform, whose materials are widely used in teaching practice[5]. In addition, the questionnaires were reviewed and administered with the support of undergraduate teaching assistants, contributing to the adequacy and clarity of the assessment instruments.

6 Conclusion

Nowadays, ChatGPT has proven valuable for a variety of purposes, including providing instant feedback and explanations. However, many skeptics argue that ChatGPT might significantly impact classroom learning and understanding, while others view these tools as transformative and disruptive technologies that can enhance education. In this context, this paper presents an experiment that analyzes the influence of ChatGPT tool usage on practical assignments in computer engineering courses, particularly those focused on data structures and algorithms. We divided a group of third-year students into two groups: one was encouraged to use ChatGPT freely on all lab assignments, and the other used it only once. The experiment evaluated some hypotheses by examining questionnaire results from post-lab sessions, final exams, and by measuring code similarity using MOSS. The main findings are as follows: (i) comparing the participants' scores on questionnaires and final exams between the group using ChatGPT and the group not using them, we found no statistically significant differences; (ii) comparing the code provided by participants using the similarity score calculated by MOSS, we identified a statistically significant difference, with higher similarity in the code from students allowed to use ChatGPT.

[4] https://www.sanfoundry.com

[5] https://www.sanfoundry.com/data-structure-questions-answers-avl-tree/ and https://www.sanfoundry.com/data-structure-questions-answers-binary-search-tree/

Based on the results, we cannot conclude that using ChatGPT during lab sessions affects learning in the ADE process; further research is required. Besides, lecturers must be aware of students' potential use of ChatGPT and design assignments with this consideration in mind. When designing assignments, lecturers should review the output from commonly used ChatGPT models to ensure they do not provide straightforward answers. Additionally, lecturers should consider creating questions that are challenging for ChatGPT to answer, such as those involving formulas or images [16]. Personalized data, such as birth dates or official document IDs, can also be used to customize questions. Despite these precautions, it is not recommended to prohibit the use of ChatGPT, as students are likely to use it regardless, as some authors have demonstrated [16, 26].

A significant limitation of this study is the need for additional replications [30]. So, for future work, we plan to apply different problems and programming languages to lab work. Here are a few ways to strengthen the validity of our conclusions. Additionally, it would be beneficial to investigate how the use of development tools affects midterm exam results. We are also interested in applying our experimental design to other courses, such as Introduction to Programming and Object-Oriented Programming, with specific adjustments. Furthermore, an empirical study to assess the development of essential skills (critical thinking, problem-solving, and other soft skills) [26] is necessary to understand the potential impact on future computer engineers. As discussed in Section 5, expanding the scope of replicated studies to include more institutions and conducting a multi-institutional, multinational study could provide a deeper understanding of integrating large language models into education. This broader perspective has the potential to yield more precise and robust outcomes compared to those presented in this study.

Artifact Availability

We have made all relevant materials used in this study publicly available. These artifacts are intended to allow other researchers to replicate our experimental procedures, validate our results, or build upon our work in future studies. The shared resources include:

- **Raw Data:** The anonymized data collected from participants during all phases of the experiment, including questionnaire responses and performance scores;
- **Statistical Analysis Scripts:** A Jupyter notebook containing all statistical analyses reported in this paper, including the tests applied, and visualizations generated;
- **Questionnaires:** The full set of multiple-choice questionnaires used in both the pretest and posttest phases, allowing for evaluation of the instruments' content and structure;
- **Lab Assignments:** The stubs and JUnit test suites provided to students during the lab sessions for implementing BST and AVL trees, which formed the basis of the experimental tasks.

These materials are hosted in an **anonymous 4 open science** repository and are freely available [6]. We encourage their use for replication studies, educational purposes, or further research into the integration of AI tools in software engineering education.

[6]https://anonymous.4open.science/r/Artifacts-CBSoft-2026-Education-9055/README.md

Acknowledgements

For partially supporting this work, we would like to thank INES.IA (National Institute of Science and Technology for Software Engineering Based on and for Artificial Intelligence) www.ines.org.br, CNPq grant 408817/2024-0. Sérgio Soares is partially supported by CNPq grant 307226/2026-3.

References

[1] Md Al-Amin, Mohammad Shazed Ali, Abdus Salam, Arif Khan, Ashraf Ali, Ahsan Ullah, Md Nur Alam, and Shamsul Kabir Chowdhury. 2024. History of generative Artificial Intelligence (AI) chatbots: past, present, and future development. *arXiv preprint arXiv:2402.05122* (2024).

[2] Sayed Erfan Arefin, Tasnia Ashrafi Heya, Hasan Al-Qudah, Ynes Ineza, and Abdul Serwadda. 2023. Unmasking the giant: A comprehensive evaluation of ChatGPT's proficiency in coding algorithms and data structures. *arXiv preprint arXiv:2307.05360* (2023).

[3] David Baidoo-Anu and Leticia Owusu Ansah. 2023. Education in the era of generative artificial intelligence (AI): Understanding the potential benefits of ChatGPT in promoting teaching and learning. *Journal of AI* 7, 1 (2023), 52–62.

[4] Kevin W Bowyer and Lawrence O Hall. 1999. Experience using" MOSS" to detect cheating on programming assignments. In *FIE'99 Frontiers in Education. 29th Annual Frontiers in Education Conference. Designing the Future of Science and Engineering Education. Conference Proceedings (IEEE Cat. No. 99CH37011*, Vol. 3. IEEE, 13B3–18.

[5] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. 2022. *Introduction to algorithms.* MIT press.

[6] Marian Daun and Jennifer Brings. 2023. How ChatGPT will change software engineering education. In *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1.* 110–116.

[7] Joanna F DeFranco, Nir Kshetri, and Jeffrey Voas. 2023. Are We Writing for Bots or Humans? *Computer* 56, 09 (2023), 13–14.

[8] Iris Delikoura, Yi R Fung, and Pan Hui. 2025. From superficial outputs to superficial learning: Risks of large language models in education. *arXiv preprint arXiv:2509.21972* (2025).

[9] Juan Dempere, Kennedy Modugu, Allam Hesham, and Lakshmana Kumar Ramasamy. 2023. The impact of ChatGPT on higher education. In *Frontiers in Education*, Vol. 8. Frontiers Media SA, 1206936.

[10] Yogesh K Dwivedi, Nir Kshetri, Laurie Hughes, Emma Louise Slade, Anand Jeyaraj, Arpan Kumar Kar, Abdullah M Baabdullah, Alex Koohang, Vishnupriya Raghavan, Manju Ahuja, et al. 2023. "So what if ChatGPT wrote it?" Multidisciplinary perspectives on opportunities, challenges and implications of generative conversational AI for research, practice and policy. *International Journal of Information Management* 71 (2023), 102642.

[11] Manuel B Garcia. 2025. Teaching and learning computer programming using ChatGPT: A rapid review of literature amid the rise of generative AI technologies. *Education and Information Technologies* (2025), 1–25.

[12] Don Gotterbarn, Keith Miller, and Simon Rogerson. 1999. Computer society and ACM approve software engineering code of ethics. *Computer* 32, 10 (1999), 84–88.

[13] Pooriya Jamie, Reyhaneh Hajihashemi, and Sharareh Alipour. 2024. Utilizing ChatGPT in a Data Structures and Algorithms Course: A Teaching Assistant's Perspective. *arXiv preprint arXiv:2410.08899* (2024).

[14] Ishika Joshi, Ritvik Budhiraja, Harshal Dev, Jahnvi Kadia, Mohammad Osama Ataullah, Sayan Mitra, Harshal D Akolekar, and Dhruv Kumar. 2024. ChatGPT in the classroom: An analysis of its strengths and weaknesses for solving undergraduate computer science questions. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1.* 625–631.

[15] Barbara Kitchenham, Lech Madeyski, and Pearl Brereton. 2019. Problems with statistical practice in human-centric software engineering experiments. In *Proceedings of the 23rd International Conference on Evaluation and Assessment in Software Engineering.* 134–143.

[16] Tomaž Kosar, Dragana Ostojić, Yu David Liu, and Marjan Mernik. 2024. Computer Science Education in ChatGPT Era: Experiences from an Experiment in a Programming Course for Novice Programmers. *Mathematics* 12, 5 (2024), 629.

[17] Richard Joseph LeBlanc, Ann Sobel, Jorge L Diaz-Herrera, and Thomas B Hilburn. 2015. *Software engineering 2014: curriculum guidelines for undergraduate degree programs in software engineering.* IEEE Computer Society.

[18] Susan M Lord, Richard A Layton, and Matthew W Ohland. 2011. Trajectories of electrical engineering and computer engineering students by race and gender. *IEEE Transactions on education* 54, 4 (2011), 610–618.

[19] Stephen MacNeil, Andrew Tran, Dan Mogil, Seth Bernstein, Erin Ross, and Ziheng Huang. 2022. Generating diverse code explanations using the gpt-3 large language model. In *Proceedings of the 2022 ACM Conference on International Computing Education Research-Volume 2*. 37–39.

[20] Leo A Meyerovich and Ariel S Rabkin. 2013. Empirical analysis of programming language adoption. In *Proceedings of the 2013 ACM SIGPLAN international conference on Object oriented programming systems languages & applications*. 1–18.

[21] Sonia Alejandrina Sotelo Muñoz, Giovanna Gutiérrez Gayoso, Alberto Caceres Huambo, Rogelio Domingo Cahuana Tapia, Jorge Layme Incaluque, Oscar Eduardo Pongo Aguila, Juan Cielo Ramírez Cajamarca, Jesus Enrique Reyes Acevedo, Herbert Victor Huaranga Rivera, and José Luis Arias-Gonzáles. 2023. Examining the impacts of ChatGPT on student motivation and engagement. *Social Space* 23, 1 (2023), 1–27.

[22] Pedro Henrique Ramos Pinto, Vitor Meneghetti Ugulino de Araujo, Cleydson de Souza Ferreira Junior, Lutero Lima Goulart, João Vitor Cardoso Beltrão, Gabriel Silva Aguiar, Samuel José Fernandes Mendes, Filipe de Lima Vaz Monteiro, and Erlon Lacerda Avelino. 2023. Assessing the Psychological Impact of Generative AI on Data Science Education: An Exploratory Study. (2023).

[23] Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. 2023. Is chatgpt a general-purpose natural language processing task solver? *arXiv preprint arXiv:2302.06476* (2023).

[24] Basit Qureshi. 2023. Exploring the use of chatgpt as a tool for learning and assessment in undergraduate computer science curriculum: Opportunities and challenges. *arXiv preprint arXiv:2304.11214* (2023).

[25] Md Mostafizer Rahman and Yutaka Watanobe. 2023. ChatGPT for education and research: Opportunities, threats, and strategies. *Applied Sciences* 13, 9 (2023), 5783.

[26] Luis M Sánchez-Ruiz, Santiago Moll-López, Adolfo Nuñez-Pérez, José Antonio Moraño-Fernández, and Erika Vega-Fleitas. 2023. ChatGPT challenges blended learning methodologies in engineering education: A case study in mathematics. *Applied Sciences* 13, 10 (2023), 6039.

[27] David J Sheskin. 2003. *Handbook of parametric and nonparametric statistical procedures*. Chapman and hall/CRC.

[28] Abdulhadi Shoufan. 2023. Can students without prior knowledge use ChatGPT to answer test questions? An empirical study. *ACM Transactions on Computing Education* 23, 4 (2023), 1–29.

[29] Rashi Shrivastava. 2022. Teachers fear ChatGPT will make cheating even easier than ever. Forbes.

[30] Forrest J Shull, Jeffrey C Carver, Sira Vegas, and Natalia Juristo. 2008. The role of replications in empirical software engineering. *Empirical software engineering* 13 (2008), 211–218.

[31] Stackoverflow. 2025. 2025 Developer Survey. https://survey.stackoverflow.co/2025/ai Last accessed 21 January 2026.

[32] C Sujatha, Jayalakshmi G Naragund, and Suvarna G Kanakaraddi. 2014. A Framework for Curriculum Design of Data Structures and Design of Algorithms Course (DSDA). In *Proceedings of the International Conference on Transformations in Engineering Education: ICTIEE 2014*. Springer, 615–615.

[33] Jessica Vitak, Nicholas Proferes, Katie Shilton, and Zahra Ashktorab. 2017. Ethics regulation in social computing research: Examining the role of institutional review boards. *Journal of Empirical Research on Human Research Ethics* 12, 5 (2017), 372–382.

[34] Leon E Winslow. 1996. Programming pedagogy—a psychological overview. *ACM Sigcse Bulletin* 28, 3 (1996), 17–22.

[35] Claes Wohlin, Per Runeson, Martin Höst, Magnus C Ohlsson, Björn Regnell, and Anders Wesslén. 2012. *Experimentation in software engineering*. Springer Science & Business Media.

[36] Ramazan Yilmaz and Fatma Gizem Karaoglan Yilmaz. 2023. Augmented intelligence in programming learning: Examining student views on the use of ChatGPT for programming learning. *Computers in Human Behavior: Artificial Humans* 1, 2 (2023), 100005.

[37] Hao Yu. 2023. Reflection on whether Chat GPT should be banned by academia from the perspective of education and teaching. *Frontiers in Psychology* 14 (2023), 1181712.

[38] Kevin KF Yuen, Dennis YW Liu, and Hong Va Leong. 2023. Competitive programming in computational thinking and problem solving education. *Computer Applications in Engineering Education* 31, 4 (2023), 850–866.