AI Generated Code Plagiarism Detection in Computer Science courses: A Literature Mapping

Archer Simmons
Texas A&M University
College Station, TX, U.S.
0009-0000-5595-9639

Maristela Holanda
Universidade de Brasília
Brasília, DF, Brazil
0000-0002-0883-2579

Christiana Chamon*
Virginia Tech
Blacksburg, VA, U.S.
0000-0003-3366-8894

Dilma Da Silva
Texas A&M University
College Station, TX, U.S.
0000-0001-6538-2888

***Abstract*—This is a full research paper. Integrity in the detection of plagiarism in students' source codes in university programming courses is a research topic for instructors and institutions seeking to improve the quality of their teaching. In particular, introductory courses such as CS1, are of paramount importance, as this is when students gain fundamental knowledge to build their future on. With the latest developments in Large Language Models (LLM) such as ChatGPT, GitHub Copilot, etc., methods of plagiarism have evolved, however methods of detection may not be capable of accurately differentiating between code generated by human and artificial intelligence (AI). In this context, this paper seeks to answer the research question: What does the current literature report on AI generated code plagiarism detection in higher education? To expand on and formulate a comprehensive answer to our research question (RQ), we have formulated six sub-questions: RQ1) How many papers were published per year by country?; RQ2) Which conferences and journals have published most papers on this subject?; RQ3) Which plagiarism detection tools were most often used prior to common AI use?; RQ4) How are educators adapting assignments to minimize the use of AI?; RQ5) Which modern methods are being deployed to specifically detect AI?; RQ6) Which data sources and languages are most prevalent in the literature? The methodology was based on a systematic literature review. Initially, we confined our search for literature to Scopus and Web of Science, however additional literature was included from Google Scholar. Inclusion criteria were applied to include documents from the years 2023 and 2024 (after the launch of ChatGPT), and only published by conferences and journals. Exclusion criteria: papers that do not focus on plagiarism and programming courses; papers that are not about the undergraduate-level; papers not written in English. We found 165 papers via Scopus and WebScience, from which the metadata were collected, resulting in 17 relevant papers selected for this work. The second step was a search in Google Scholar, where we analyzed 200 documents from 2023 (100 relevant documents) and 2024 (100 relevant documents). We used the same inclusion and exclusion criteria, however, we included the ArXiv papers, and found 9 more papers. Following this process, we have identified 26 papers to include in this literary mapping. In this paper we present the answers to these research questions and discussions about this research topic.**

***Index Terms* — programming, plagiarism detection, code, AI, academia, education, similarity, mapping**

*This research project was carried out while the author was at Texas A&M University

I. Introduction

Plagiarism involves appropriating another person's intellectual work and presenting it as if it were one's own [1]. This behavior occurs across multiple disciplines, such as music, literature, arts, and academic environments. As a byproduct of the modern day Internet, the rise of web-services which make it easier for students to commit plagiarism in an academic environment are easily accessible and widely used. Since the release of ChatGPT in 2022, AI (artificial intelligence) chatbots have exploded in popularity, and they are continuously evolving in their sophistication, presenting countless new opportunities and challenges, one of which is the rise of plagiarism utilizing AI ChatBots. Many recent studies have addressed the topic of generative artificial intelligence (AI) and plagiarism in higher education [2]–[7].

A new challenge that has arisen is reliably detecting AI-assisted plagiarism, specifically in programming courses. Unlike other subjects which have tools readily available to detect the use of AI, detection of AI-generated code in student submissions is significantly more challenging as existing code-plagiarism detection software is not capable of reliably detecting AI-generated code [8]. This issue should command the attention of educators across the globe, as it potentially threatens the integrity and potentially the reputation of academic institutions. Unlike traditional code-plagiarism which involved copying code found online or provided by a peer, due to the nondeterministic nature of large language model (LLM) based chatbots, new methods of detection are needed as the existing software is inadequate. Additionally, literature regarding detection of AI generated code in students´ work is limited, however new promising methods of detection as well as methods of thwarting the use of AI by students have been proposed in the available literature. Ironically, the same technology that has caused this major issue is the same technology we will most likely see working to minimize the issue.

In this paper, we will answer six research questions by systematically mapping research literature to address the main research question: What does the current literature suggest regarding AI-generated code plagiarism detection in higher education?

Our search for literature was localized to Scopus, Web of

Science, and Google Scholar filtered utilizing inclusion and exclusion criteria, resulting in a set of 26 papers that we analyzed to answer the aforementioned question.

Remaining sections within this literary mapping are organized as follows: Section II specifies the research questions of this literature mapping, Section III describes the literature selection process of relevant papers, in Section IV, the results of our research questions are presented, Section V discusses the findings, Section VI shows the limitations of this paper, and our conclusions are presented in Section VII.

II. Research Questions

In this paper, we utilized a systematic literature mapping process, based on [9], involving four stages:

i. Define the research questions.
ii. Select papers relevant to this research.
iii. Analyze the selected research papers.
iv. Answer each of the research questions.

Our main research question is as follows: What does the current literature report on AI generated code plagiarism detection in higher education? To expand on and formulate a comprehensive answer to our research question, we formulated six sub-questions:

- RQ1: How many papers were published per year by country?
- RQ2: Which conferences and journals have published most papers on this subject?
- RQ3: Which plagiarism detection tools were most often used prior to common AI use?
- RQ4: How are educators adapting assignments to minimize the use of AI?
- RQ5: Which modern methods are being deployed to specifically detect AI?
- RQ6: Which data sources and languages are most prevalent in the literature?

III. Selecting the relevant papers

When conducting academic research, the literature selection stage, the process by which relevant literature is selected for inclusion, is integral to a comprehensive literary mapping. To begin the process, (a) database(s) must be selected. Next, a search string is constructed such that it maximizes the reach of the search without over-saturating the selection pool with literature irrelevant to the research. Lastly, inclusion and exclusion criteria further refine the search, producing a more significant selection pool.

When constructing our search string, care was taken to select keywords which were not overly broad, which would result in a saturation of literature which would not be selected due to irrelevance to the research being conducted. Similarly, keywords that are too specific would result in unintentional exclusion of relevant papers, which should be avoided. The following search string was our initial search string:

(code OR algorithm OR algorithms OR ‘programming OR “Computer Science Education”) AND (plagiarism OR “similarity detection” OR cheat OR cheating)

This search string was used in a previous research paper regarding code-plagiarism in computer science courses [12], however due to the dramatic change to the subject given the introduction of LLM chat-bots, the resulting literature selection pool was too large, containing hundreds of papers which made no mention of AI. To remedy this issue, additional keywords were added to focus the results pool on detection of plagiarism utilizing AI. These keywords include the names of popular online AI chat bots such as ChatGPT, GitHub Copilot, Bard, and other terms generally related to AI chat bots. These additions resulted in the following search string:

(programming OR coding OR “computer science” OR code OR algorithm OR algorithms)
AND
(“academic dishonesty” OR “unethical behavior” OR “integrity in academia” OR plagiarism OR cheat OR cheating OR “academic misconduct” OR “code similarity”)
AND
(“AI-generated code” OR LLM OR ChatGPT OR OpenAI OR Bard OR Copilot OR “AI in education” OR “code generation” OR “Artificial Intelligence”)

Initially, we intended to include databases such as Scopus and Web of Science, though due to how recently this issue arose and therefore the low volume of literature regarding the research topic, Google Scholar was added to our inclusion.

Once all literature resulting from our search string was indexed, we applied inclusion criteria (IC) to further refine our results, as only literature after the chatbots were released is of importance for this literary review:

- IC1: Papers must be written in English.
- IC2: Papers must be published between 2023 and 2024.
- IC3: Papers must either be classified as a ”conference paper” or ”journal article”.

The exclusion criteria (EC) were:

- EC1: Papers with fewer than 4 pages.
- EC2: Papers are out of context for this literature mapping.
- EC3: Papers not classified as a ”conference paper” or ”journal article”.

We collected their metadata and put them in a spreadsheet. The metadata comprised title, authors, source, abstract, year, document type, source title, and page count. Figure 1 presents the process to find relevant papers. We found 185 papers on Scopus from searches carried out in February and April 2024 (102 and 83 papers respectively), while in Web of Science 76 papers were found from searches in February (63 papers) and April (13 papers). With regard to Google Scholar a total of 200 relevant papers was found, 100 in each of the years 2023 and 2024. After that, the process consists of eliminating all papers that are clearly unrelated to our study based on the metadata, we applied the EC1, EC2 and EC3.

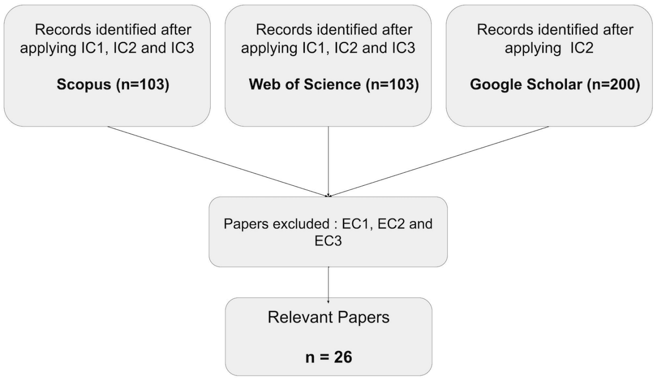

Fig. 1. Summary of inclusion/exclusion criteria in the research stages.

Following our aforementioned refining process, which is further explored in Table I, a total of 26 papers were selected for inclusion in the literary review. These results are presented in the table below by the number of papers selected per database.

TABLE I
SEARCH RESULTS

Source	No.	Papers
Scopus	15	[13] [14] [15] [16] [8] [17] [27] [28] [29] [30] [31] [32] [33] [34] [36]
Google Scholar	9	[19] [20] [21] [22] [23] [24] [25] [26] [37]
Web of Science	2	[18] [35]
Final Results	26	

IV. Results and Analysis

A. RQ1: How many papers were published per year by country?

Provided below, Figure 2 shows the distribution of papers published by year. As a result of the only recent deployment of AI chatbots in November 2022, our relevant papers were published within a two-year range (2023 and 2024). From 2023 to 2024, the volume of literature published increased as interest in the subject increased.

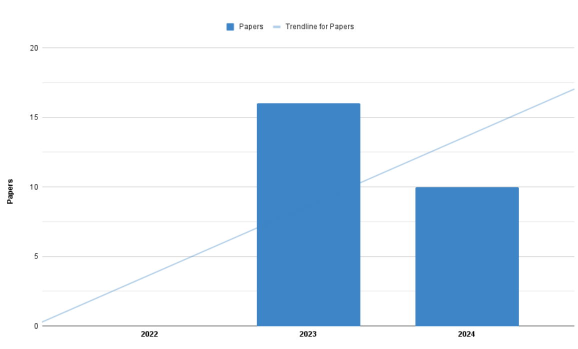

Fig. 2. Number of papers published per year

The sudden rise of papers from 2023 to 2024 is most likely due to the November 2022 release and subsequent rise in popularity of ChatGPT 3.5 to the public, thus bringing it to the attention of researchers and educators alike.

Additionally, we identified publications from 16 countries in 4 different continents. Figure 3 shows a map indicating the distribution of these countries.

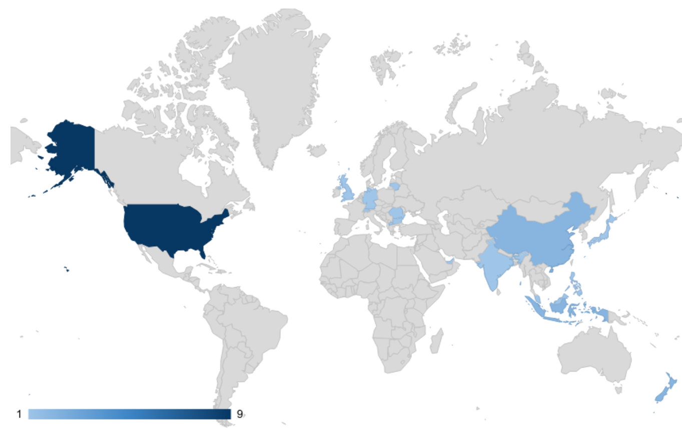

Fig. 3. Distribution of the papers by countries.

The numbers of papers produced by the 13 source countries is shown in Table II. Of the identified countries, the United States published the most, with Indonesia, China, and New Zealand as runner ups for frequency of publication regarding this topic.

TABLE II
SELECTED PAPERS PUBLISHED PER COUNTRY

Country	No.	Papers
United States	9	[13] [15] [8] [17] [31] [32] [33] [36] [37]
Indonesia	2	[26] [21]
China	2	[28] [30]
New Zealand	2	[27] [33]
Bulgaria	1	[14]
Macedonia	1	[16]
Romania	1	[18]
Germany	1	[19]
India	1	[22]
United Kingdom	1	[23]
Japan	1	[24]
Malaysia	1	[25]
Switzerland	1	[20]
Philippines	1	[29]
UAE	1	[34]
Lithuania	1	[35]

B. RQ2: Which conferences and journals have published most papers on this subject?

The literature included in our selection originated from 10 unique journals and 12 unique conferences. As shown in both tables, Table III and Table IV, the diversity of publication source is high as a majority of relevant papers were each pulled from unique journals and conferences.

TABLE III
JOURNALS WITH THE MOST PUBLICATIONS

Journal	No.	Papers
ArXiv	2	[22] [25]
Service Industries Journal	1	[14]
IEEE Intelligent Systems	1	[15]
International Journal of Human-Computer Interaction	1	[18]
ACM Transactions on Computing Education	1	[23]
Analytics	1	[24]
Journal of Information Science	1	[34]
Education & Information Technologies	1	[26] [35]
Decision Sciences Journal of Innovative Education	1	[37]

Similar to the diverse nature of the journal sources, conference sources were equally as diverse, with SIGCSE 2024 proceeding as our largest single source for our selected literature.

TABLE IV
CONFERENCES WITH THE MOST PUBLICATIONS

Conference	No.	Papers
SIGSCE	3	[32] [33] [36]
ICER	2	[13] [17]
Frontiers in Education (FIE)	2	[27] [31]
Procedia Computer Science	1	[28]
IEEE ICSPC	1	[29]
EMNLP	1	[30]
HEAd	1	[16]
AIEDLLM	1	[8]
IEEE International Conference on Software Engineering Education & Training	1	[19]
Frontiers in Software Engineering Education: International Workshop	1	[20]
IEEE International Conference on Teaching, Assessment & Learning for Engineering	1	[21]

C. RQ3: Which plagiarism detection tools were most often used prior to common AI use?

In an effort to understand detection of AI, it is important to first understand which traditional tools are being utilized by educators, and that these may no longer be reliable for detection of academic dishonesty when analysing AI-generated code due to its nondeterministic nature. The following papers shed light on this: [8], [15], [19], [20], [22], [23], [28], [32]–[34], [37].

TABLE V
TRADITIONAL DETECTION ALGORITHMS

Algorithm	Software	Deterministic	Papers
Rabin-Karp	MOSS, JPLAG	YES	[33], [32] [33] [28] [16] [34]
Support Vector Machine	NA	NO	[8]
XGBoost	NA	NO	[8]

Out of the traditional methods of detection listed, similarity checking algorithms such as MOSS appeared to be the most common by far. However, some machine learning models were used in the past, which are now unreliable in the detection of AI use. Note that this nondeterministic nature of a detection algorithm implies that, rather than a static code match-rate, these methods implore probabilistic analysis by leveraging modern machine learning and artificial intelligence to produce a probability of the code being AI-generated. However, those listed in Table VII are outdated and less reliable in detecting code generated by newer AI models [8].

D. RQ4: How are educators adapting assignments to minimize the use of AI?

In Table VI, assignment modifications and ways being proposed by a number of authors as methods to minimize the use of AI chat bots in student assignments are listed. These methods attempt to make it more difficult to utilize AI on their assignments, leading to less usage by students. Each of the following methods leverages one or more of AI chat bots' own weaknesses against it.

TABLE VI
ASSIGNMENT ADAPTATIONS

Paper	Method
[17] [19] [21] [23] [31] [32] [36]	Local or Complex Context
[19] [17]	Non-Virtual Assessments
[19]	Comprehension-Based Requirements

The following subsections provide brief descriptions of each method in Table VI proposed to professors as potentially effective methods to minimize the utilization of AI by students on their assignments. Note that, while many individual methods were suggested in the literature, we have grouped many independent methods into common categories.

1) Local or Complex Context: Local and complex contexts refer to contextual information provided by instructors which is specific to the course, is not text-based, is text-based though too lengthy for AI's limited context window [21], or that requires knowledge only available in the course assigning the assignment. This also incorporates information which is openly provided and in text, however the context is intentionally made too long for publicly available chatbots. One example is requiring students to utilize custom libraries written by the professor potentially comprised of hundreds of lines of code [17], though inversely, this may entail providing shortened problem statements which would open the assignment up for varying interpretations to outside tools such as chatbots [19]. Lastly, the use of images and/or videos in problem statements may also be a viable option as many of the most popular chatbots interact only via text rather than images, meaning only the student may interpret the visual media [17] [19] [23].

2) Non-Virtual Assignments: Non-virtual assignments is a brute-force approach to reducing the use of AI chatbots in courses as, to interact with chatbots in the first place, students must have access to a virtual device. Requiring assignments such as exams to be done on paper only, assures that students cannot and do not have access to nor utilize chatbots [17]. The effectiveness of this method can be bolstered by modifying the course syllabus so that paper-based assignments such as exams are weighted significantly higher than other hybrid or virtual assignments; by assigning heavier weights to paper assignments, a given student's grade will more closely reflect the student's competency in the course without the assistance of chat bots [17].

3) Comprehension Assessments: Comprehension assessments are requirements added to assignments which require in-depth knowledge of the completed assignment to then produce documents or media summarizing the assignment, or that can be subjectively assessed. For example, a typical programming assignment could include an added requirement that a student write clear documentation for their solution code or explain in detail the process they followed from start to finish to reach their solution [19]. Other metrics can be assessed as well including the quality or efficiency of the submitted code, or adding test-cases for submitted code which are hidden and/or dynamic [19].

E. RQ5: Which modern methods are being deployed to specifically detect AI?

Given the unreliability of traditional detection methods, though the outcomes of nondeterministic leveraging machine learning are promising, more reliable identification AI-generated code may be possible with modern detection algorithms, which are themselves powered by AI [8].

TABLE VII
MODERN DETECTION ALGORITHMS

Algorithm	**Software**	**Deter.**	**Papers**
Deep Learning Models	code2vec	NO	[22] [33] [34] [35]
Abstract Syntax Tree	NA	NO	[8] [33]
Text Classifier	GPTZero	NO	[22] [25]
Hybrid	CodeQuiry	NO	[20]

Out of the methods utilized in our selected literature, deep learning models were the most common. The reason for the focus on non-deterministic methods is that AI-generated code is not consistent in its output, meaning the same prompt may yield different code. Due to this, utilizing traditional detection tools is no longer reliable in detecting plagiarism, specifically, code generated by chatbots. To counteract this, the most promising methods suggested for detection leverages machine learning and artificial intelligence against itself. By training models on student-submissions and AI-generated submissions, the prediction models yielded excellent results of greater than 90% accuracy rate, which is the highest accuracy rate discussed in the literature [8].

F. RQ6: Which data sources and languages are most prevalent in the literature?

Following our literature selection process and subsequent analysis of the relevant literature, it was noted that around 35% of the studies were language-agnostic; no particular languages were mentioned in the text [14]–[16], [35]. The remaining literature specified a minimum of one language used by the students, each of which were analysed for a deeper insight. It was noted that some programming languages were more common than other languages. Given the significantly higher frequency, this may suggest that the language is more widely used in programming courses or in experiments.

The frequency at which the identified programming languages were utilized in the selected literature is shown in Table VIII.

TABLE VIII
LANGUAGE FREQUENCY

Language	**No.**	**Papers**
Python	11	[13] [17] [20] [23] [24] [36] [25] [26] [27] [34] [37]
Java	4	[8] [19] [24] [33]
C/C++	2	[31] [32]
PHP	1	[21]
JavaScript	1	[21]
Assembly	1	[31]

As shown in Table VIII, the most common language, Python, was nearly three times more frequently used than the runner-up, Java. This is likely due to the low-verbosity of the Python programming language, which is appealing to students given the context of introductory courses or boot-camps.

In the following table, Table IX, the most common sources for plagiarism detection experiments is listed per the literature.

TABLE IX
DATA SOURCES

Data Source	**No.**	**Papers**
Student Submissions	15	[16] [8] [17] [20] [21] [23] [24] [26] [29] [30] [32] [33] [35] [37] [27]
GPT/BARD	4	[13] [20] [22] [34]
LeetCode	1	[21] [25]

LeetCode is a website popular with software engineers used for preparing for technical interviews; they offer a wide array of prompts that test programming skills.

The data collected above suggests that for this sub-field of plagiarism detection, the main data sources are student submissions and GPT-generated code. This is to be expected as the objective is to develop the ability to distinguish AI-generated code from student code; as ChatGPT is the most popular AI chat bot available, it is an excellent data source as it is where many students will turn to for AI-assistance.

Lastly, of the data compiled from the literature selected, every paper which specified a specific course relevant to the research stated the data used was collected from introductory computer science courses, a vast majority of which utilized the Python programming language. This may imply that, though the newly proposed machine learning models for detection are successful in detecting AI-generated code in introductory courses, no information was collected regarding its efficacy in higher-level courses. Nonetheless, the results and suggestions from each paper are integral to the progression of detection, particularly for AI-generated code.

V. Discussion

This section presents a discussion regarding the outcomes of our research questions.

Since their major debut in late 2022 and subsequent rise to popularity, partially due to their ability to solve assignments for students, AI chat bots have earned their spot in the ranks of largest threats to the academic integrity of academic institutions. Given their sudden appearance, academic literature regarding specific detection of AI use, which requires a new set of tools, is sparse, and common-practice policies or procedures for educational institutions have yet to be implemented.

However, from the existing literature, many methods of detecting AI-generated code and assignment design methods have been proposed for professors around the globe to implement in their own lectures. Using the increase of publications from 2023 to 2024 as a prediction metric, an increasing number of publications may soon be available to add valuable information to professors in the battle against AI plagiarism, with the United States leading the way in the sub-field.

Utilizing the information shared in the available literature, we may also begin to see commercial media and tools become available to educators to increase accessibility to both detection software and assignment-structuring methods.

While assignment-structuring appears to be the most reliable tactic for minimizing academic dishonesty stemming from chat bot usage, nevertheless with the rate at which chat bots are advancing, the efficacy of each method, aside from non-virtual assessments, may diminish. For virtual educators, it is vital that detection software continues to advance at the same rate, if not a higher rate, than the publicly available chat bots.

VI. Limitations

This paper is a study of publications from conferences and journals. Any initiatives not disseminated through the publication of academic papers are excluded. Furthermore, a common threat in systematic mapping is inadequate coverage of papers on the topic area. To ensure the inclusion of relevant documents, the definition of the search string was based on a previous paper [12]. As the research topic is new, the articles may not yet be indexed in the academic databases we use in literature mapping.

VII. Conclusion

In this paper, we compiled and discussed a literature mapping relevant to the detection of AI-generated code plagiarism in academic institutions.

From the relevant literature, it is clear that educators are currently struggling to detect and prevent students' use of AI chat bots due to chat bots being a recent technology; when used by students for these malicious reasons, chat bots become a threat to the academic integrity of any institution.

It is abundantly clear that innovation in this sub-field is integral to the future of traditionally operated higher education institutions. The most promising approach is using machine learning and AI models to assign probabilities based on learned metrics per student submission in an effort to detect academically dishonest submissions.

References

[1] Helgesson, G., & Eriksson, S. (2015). Plagiarism in research. Medicine, Health Care and Philosophy, 18, 91-101.

[2] Gill, S. S., Xu, M., Patros, P., Wu, H., Kaur, R., Kaur, K., ... & Buyya, R. (2024). Transformative effects of ChatGPT on modern education: Emerging Era of AI Chatbots. Internet of Things and Cyber-Physical Systems, 4, 19-23.

[3] Wang, N., Wang, X., & Su, Y. S. (2024). Critical analysis of the technological affordances, challenges and future directions of Generative AI in education: a systematic review. Asia Pacific Journal of Education, 1-17.

[4] Riyadi, A. R., Syaripudin, T., Hendriani, A., Sutinah, C., & Sirojudin, M. A. (2024). ChatGPT and Education: A Scopus Bibliometric Analysis. In International Conference on Teaching, Learning and Technology (ICTLT 2023) (pp. 264-272). Atlantis Press.

[5] Rahman, M. M., & Watanobe, Y. (2023). ChatGPT for education and research: Opportunities, threats, and strategies. Applied Sciences, 13(9), 5783.

[6] Rajabi, P., Taghipour, P., Cukierman, D., & Doleck, T. (2023). Exploring ChatGPT's impact on post-secondary education: A qualitative study. In Proceedings of the 25th Western Canadian Conference on Computing Education (pp. 1-6).

[7] Meyer, J. G., Urbanowicz, R. J., Martin, P. C., O'Connor, K., Li, R., Peng, P. C., ... & Moore, J. H. (2023). ChatGPT and large language models in academia: opportunities and challenges. BioData Mining, 16(1), 20.

[8] M. Hoq et al., "Detecting ChatGPT-Generated code submissions in a CS1 course using machine learning models," in ACM Digital Library, Mar. 07, 2024. doi: 10.1145/3626252.3630826.

[9] B. Kitchenham and S. Charters. 2007. Guidelines for Performing Systematic Literature Reviews in Software Engineering. Technical Report EBSE 2007-001. KeeleUniversity and Durham University.

[10] Steponenaite, A., & Barakat, B. (2023). Plagiarism in AI empowered world. In International Conference on Human-Computer Interaction (pp. 434-442). Cham: Springer Nature Switzerland.

[11] Lancaster, T. (2023). Academic integrity for computer science instructors. In Higher education computer science: A manual of practical approaches (pp. 71-84). Cham: Springer International Publishing.

[12] Aniceto, R. C., Holanda, M., Castanho, C., & Da Silva, D. (2021). Source code plagiarism detection in an educational context: A literature mapping. In 2021 IEEE Frontiers in Education Conference (FIE) (pp. 1-9). IEEE.

[13] J. Šavelka, A. Agarwal, M. An, C. Bogart, and M. Sakr, "Thrilled by your progress! Large language models (GPT-4) no longer struggle to pass assessments in higher education programming courses," ACM Digital Library, Aug. 2023, doi: 10.1145/3568813.3600142

[14] S. Ivanov, "The dark side of artificial intelligence in higher education," The Service Industries Journal, vol. 43, no. 15–16, pp. 1055–1082, Sep. 2023, doi: 10.1080/02642069.2023.2258799.

[15] H. Ibrahim, R. Asim, F. Zaffar, T. Rahwan, and Y. Zaki, "Rethinking homework in the age of artificial intelligence," IEEE Intelligent Systems, vol. 38, no. 2, pp. 24–27, Mar. 2023, doi: 10.1109/mis.2023.3255599.

[16] K. Zdravkova, "Evolution of academic dishonesty in computer science courses," in International Conference on Higher Education Advances (HEAd), Jun. 19, 2023. doi: 10.4995/head23.2023.16081.

[17] S. Lau and P. J. Guo, From "Ban it till We Understand It" to "Resistance is Futile": How university programming instructors plan to adapt as more students use AI code generation and explanation tools such as ChatGPT and GitHub Copilot, vol. 1. 2023. doi: 10.1145/3568813.3600138.

[18] M.-D. Popovici, "ChatGPT in the Classroom. Exploring its potential and limitations in a functional programming course," International Journal of Human-computer Interaction, pp. 1–12, Oct. 2023, doi: 10.1080/10447318.2023.2269006.

[19] J. Berrezueta-Guzmán and S. Krusche, "Recommendations to create programming exercises to overcome ChatGPT," in IEEE Xplore, Tokyo, Japan, Aug. 01, 2023. doi: 10.1109/cseet58097.2023.00031.

[20] J. Kotovich and M. Oriol, "Is ChatGPT 3 safe for students?," in Lecture notes in computer science, 2023, pp. 100–107. doi: 10.1007/978-3-031-48639-5_8.

[21] O. Karnalim, H. Toba, M. C. Johan, E. D. Handoyo, Y. Setiawan, and J. A. Luwia, ”Plagiarism and AI assistance Misuse in web programming: Unfair benefits and characteristics,” arXiv (Cornell University), Oct. 2023, doi: 10.48550/arxiv.2310.20104.

[22] S. Sharma and B. Sodhi, ”Calculating originality of LLM assisted source code,” arXiv (Cornell University), Jan. 2023, doi: 10.48550/arxiv.2307.04492.

[23] K. Waugh, M. Slaymaker, M. Petre, J. Woodthorpe, and D. Gooch, “Bob or Bot: Exploring ChatGPT’s answers to University Computer Science Assessment,” ACM Transactions on Computing Education, Nov. 2023, doi: 10.1145/3633287.

[24] E. E. Htet et al., ”Code Plagiarism checking function and its application for code writing problem in Java Programming Learning Assistant System,” Analytics (Basel), vol. 3, no. 1, pp. 46–62, doi: 10.3390/analytics3010004.

[25] W. Pan et al., ”Assessing AI Detectors in Identifying AI-Generated Code: Implications for education,” arXiv (Cornell University), Jan. 2024, doi: 10.48550/arxiv.2401.03676.

[26] O. Karnalim, H. Toba, and M. C. Johan, ”Detecting AI assisted submissions in introductory programming via code anomaly,” Education and Information Technologies, Feb. 2024, doi: 10.1007/s10639-024-12520-6.

[27] H. Hickman, P. McKeown, and T. Bell, Beyond question shuffling: Randomization techniques in programming assessment. IEEE, 2023. doi: 10.1109/fie58773.2023.10342976.

[28] Z. Li, J. Zhu, X. Cheng, and Q. Lu, “Optimized Algorithm Design for Text similarity Detection Based on Artificial Intelligence and Natural Language Processing,” Procedia Computer Science, vol. 228, pp. 195–202, Jan. 2023, doi: 10.1016/j.procs.2023.11.023.

[29] J. R. Padilla, M. D. Montefalcon, and A. A. Hernandez, Language AI in Programming: A Case Study of ChatGPT in Higher Eduation Using Natural Language Processing. IEEE, 2023. doi: 10.1109/icspc59664.2023.10420194.

[30] B. Zhu et al., Beat LLMs at their own game: Zero-Shot LLM-Generated Text Detection via Querying ChatGPT. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.463.

[31] M. M. Morsy, A. Farraj, and D. Reavis, On the challenges and opportunities of using ChatGPT in academia. IEEE, 2023. doi: 10.1109/fie58773.2023.10343189.

[32] F. Vahid, A. Pang, and B. Denzler, Towards comprehensive metrics for programming cheat detection. Association for Computing Machinery, Inc, 2024. doi: 10.1145/3626252.3630951.

[33] M. Hoq et al., Detecting ChatGPT-Generated code submissions in a CS1 course using machine learning models. Association for Computing Machinery, Inc, 2024. doi: 10.1145/3626252.3630826.

[34] K. Hayawi, S. Shahriar, and S. S. Mathew, “The imitation game: Detecting human and AI-generated texts in the era of ChatGPT and BARD,” Journal of Information Science, Feb. 2024, doi: 10.1177/01655515241227531.

[35] P. Stefanovič, B. Pliuskuvienė, U. Radvilaitė, and S. Ramanauskaitė, “Machine learning model for chatGPT usage detection in students’ answers to open-ended questions: Case of Lithuanian language,” Education and Information Technologies, Mar. 2024, doi: 10.1007/s10639-024-12589-z.

[36] D. Cambaz and X. Zhang, Use of AI-driven code generation models in teaching and learning programming: A Systematic Literature review. 2024. doi: 10.1145/3626252.3630958.

[37] M. E. Ellis, K. M. Casey, and G. R. Hill, “ChatGPT and Python programming homework,” Decision Sciences Journal of Innovative Education, vol. 22, no. 2, pp. 74–87, Jan. 2024, doi: 10.1111/dsji.12306.