Academia and Industry Synergy: Addressing Integrity Challenge in Programming Education

Rina Azoulay[1], Tirza Hirst[1] and Shulamit Reches[2]
[1]*Department of Computer Science, Jerusalem College of Technology, Jerusalem, Israel*
[2]*Department of Mathematics, Jerusalem College of Technology, Jerusalem, Israel*

Keywords: ChatGPT, Large Language Models, Computer Science Education, Plagiarism, Integrity, LLMs.

Abstract: This research addresses the profound challenges presented by sophisticated large language models (LLMs) like ChatGPT, especially in the context of educational settings, focusing on computer science and programming instruction. State of the art LLMs are capable of generating solutions for standard exercises that are assigned to students to bolster their analytical and programming skills. However, the ease of using AI to generate programming solutions poses a risk to the educational process and skill development, as it may lead students to depend on these solutions instead of engaging in their own problem-solving efforts. Our study suggests collaborative methods involving computer science educators and AI developers to provide evaluators with tools to distinguish between code produced by ChatGPT and code genuinely created by students. We propose a novel steganography-based technique for watermarking AI-generated code. By implementing this comprehensive strategy and effectively utilizing such technology through the combined efforts of educators, course administrators, and partnerships with AI developers, we believe it is possible to preserve the integrity of programming education in an age increasingly influenced by LLMs capable of generating code.

1 INTRODUCTION

Large language models (LLMs), such as ChatGPT, can be a valuable resource for programmers, especially those looking for assistance, inspiration, or seeking clarification on programming concepts. They are trained on a vast amount of programming-related text, which enables them to understand and produce code in various programming languages. They can assist with tasks such as identifying syntax errors, suggesting code corrections, or explaining language-specific features.

As a result, LLMs' ability to perform high-level cognitive tasks and produce human-like text has raised concerns about their potential role in academic dishonesty (Susnjak, 2022).

One of the significant challenges that has emerged as a result of LLMs is their ability to provide solutions for programming tasks that students traditionally tackle to sharpen their analytical and writing skills. In particular, ChatGPT and other tools can assist in the software development process, from code generation to optimization (Azaria et al., 2023).

While these tools offer assistance in software development, from coding to optimization, their use in educational settings can be problematic. In programming education, the aim of programming tasks is to enable students to practice problem-solving and programming, to enhance their analytical and programming skills. Given LLMs' ability to produce code solutions, there is a considerable temptation for students to utilize these ready-made solutions instead of engaging in self-practice and skill development.

Relying on AI-generated solutions could hinder the student's learning process and obstruct the crucial development of their skills. Furthermore, LLMs can produce erroneous or unsuitable outputs, a phenomenon known as 'hallucination'. Reliance on these tools can lead to students absorbing incorrect information, developing a flawed understanding of the subject matter, and adopting approaches unsuited to the task at hand, owing to this propensity of LLMs.

The challenge of ensuring integrity in academic assignments given the current AI abilities occupies the best teachers and lecturers around the world. In this paper, we propose a comprehensive set of strategies, designed to ensure integrity in programming tasks. To address plagiarism concerns, we suggest a collaborative method involving educators, assignment creators, and AI companies, focused on

Azoulay, R., Hirst, T. and Reches, S.
Academia and Industry Synergy: Addressing Integrity Challenge in Programming Education.
DOI: 10.5220/0012451000003636

In *Proceedings of the 16th International Conference on Agents and Artificial Intelligence (ICAART 2024) - Volume 3*, pages 1135-1143
ISBN: 978-989-758-680-4; ISSN: 2184-433X

a steganography-based method (Singh et al., 2009). A steganography-based method refers to a technique where information is hidden within another medium to prevent its detection. In the context of AI-generated code, we suggest embedding hidden watermarks within the text of the code to clearly indicate its AI origin. These concealed markers enable teachers and instructors to swiftly detect plagiarism, thereby ensuring accountability by visibly attributing the AI-created content to its source.

By applying this method, we aspire to construct a balanced and sustainable ecosystem of AI use within the educational sphere, effectively distinguishing between AI-generated snippets and student-crafted code to flag instances of academic dishonesty.

A major point for deliberation is what might motivate the developers of AI tools to cooperate in identifying code fragments produced by their systems. Nevertheless, we advocate establishing legal boundaries and obligations that require AI companies to adhere to specific criteria, ensuring responsible and ethical AI use in the educational domain. With the global momentum towards crafting regulations for the ethical and safe application of AI, it is reasonable to anticipate the emergence of such rules in this field. These regulations could specifically include offering special licenses to lecturers and teachers, allowing them to use the AI tool and request the implementation of steganography-based techniques. Under these regulatory measures, any company entering this market would need to adhere to these stipulations and embed identifiable markers in their code. Additionally, within the context of group licenses for academic institutions using commercial LLM tools, these tools would be required to comply with these regulatory standards. This requirement could incentivize commercial entities to meet these standards and integrate watermarks into their code, driven by financial motives.

In our study, we explore existing literature on text-based steganography. We then examine different methods for integrating these techniques into AI-generated code. Additionally, we propose communication protocols that facilitate collaboration between teachers and AI systems, aiming to produce the required encoded outputs.

The remainder of this paper is organized as follows: Section 2 presents a literature review on academic integrity in the context of AI tools. Section 3 provides an overview of text steganography and its use in identifying AI-generated content to prevent plagiarism. Section 4 outlines various steganographic methods for embedding unique markers in automatically generated code. Section 5 describes a collaborative approach between educators and AI developers to utilize steganography for detecting AI-generated content. Finally, the paper wraps up with Section 6, where we present our conclusions and suggest areas for further study.

2 RELATED WORK

We proceed by offering a detailed overview of various strategies and recommendations aimed at upholding academic integrity, particularly in light of the capability of LLMs to produce intricate content. When ChatGPT was unveiled as the inaugural LLM for public use, it spurred widespread discourse both among the general public and specifically within educational institutions. The dialogue often centered on its appropriate utilization and its broader societal impact.

The integration of ChatGPT into student education brings with it a multitude of opportunities and challenges. Neumann et al. (Neumann et al., 2023) focuses on articles addressing the impact of ChatGPT on higher education, specifically in the areas of software engineering and scientific writing. The paper recommends utilizing plagiarism checkers and AI detection tools, or manually examining the texts for ChatGPT fingerprints. Additionally, it suggests implementing an oral examination or requiring documentation of the examination process. Similarly, the review (Lo, 2023) on the impact of ChatGPT's on education, reveals the capabilities of it across subjects (strong in economics, moderate in programming, weak in math), and suggests that schools adopt assessment methods, update policies, train instructors, and educate students to effectively integrate ChatGPT.

Excessive reliance on AI may have lasting impacts and potentially undermine the professional development of upcoming generations. The study referenced in (Mijwil et al., 2023) discusses concerns about the next generation relying solely on AI to complete tasks without putting in effort, while emphasizing the importance of educating them about the limitations and potential biases of AI and how to evaluate the information it provides. The paper's main conclusion is that artificial intelligence applications like ChatGPT function as tools to support human work rather than replacing it entirely. While they can assist in task completion and enhance the quality of writing, they cannot completely replace human expertise in writing and critical thinking.

As highlighted in our introduction, the impact on computer science education holds unique significance. This topic is further explored by Quersi (Qureshi, 2023), who delves into the integration of

ChatGPT in undergraduate computer science curricula, specifically focusing on foundational programming concepts. An experiment was organized with two distinct groups of students, divided into teams, tasked with solving programming assignments: one group had access only to textbooks and notes with no internet, while the other group was equipped with ChatGPT. As it turned out, the teams using ChatGPT achieved higher average grades compared to the group that did not use it across all programming tasks. Nonetheless, they also spent more time grappling with the most complex problems. Finally, advantages and disadvantages of ChatGPT in teaching computer science are discussed, and various strategies are suggested to avoid misuse of ChatGPT by programming students, including the use of automated tools to detect plagiarism.

However, detecting AI-generated content remains a formidable challenge. While numerous tools have been developed, their efficacy is yet to be fully realized. In a study conducted by Khalil et al. (Khalil and Er, 2023), the aim was to explore the originality of content generated by ChatGPT. To accomplish this, the researchers employed two popular plagiarism detection tools to assess the originality of 50 essays produced by ChatGPT on various topics. The findings of the study manifest that ChatGPT possesses a significant potential to generate sophisticated and intricate text outputs, largely eluding detection by plagiarism-checking software.

To address complexities involved in identifying and mitigating academic dishonesty, Cotton et al. (Cotton et al., 2023) propose the establishment of policies and procedures, provision of training and support, and utilization of diverse methods to detect and prevent cheating. In addition, they suggest that teachers may consider mandating a written declaration from students asserting the originality of their work. However, such a declaration may not be genuine in actual situations. The question is what can be done to encourage students to submit independent work.

Considering foreign language studies, Perkins (Perkins, 2023) highlights the use of LLMs by foreign Language (EFL) learners, including potential assistance in digital writing and beyond, and delves into the concerns regarding academic integrity associated with students' use of these tools. He suggests that the use of LLMs should not be considered plagiarism, provided that students clearly disclose their use of the technology in their submissions. Moreover, there are legitimate uses of these tools in student education. He concludes that the determination of whether a specific use of LLMs by students is deemed academic misconduct should be based on the academic integrity policies of the institution. These policies must be updated to reflect how these tools will be utilized in upcoming educational settings.

We proceed by describing some studies suggesting how to incorporate the appropriate use of LLMs in education while handling its challenges and limitation. Kasneci et al. (Kasneci et al., 2023) caution against over-reliance, emphasizing the importance of recognizing LLM limitations and promoting teacher training. The authors advocate the use of LLMs as supplementary tools alongside other educational resources, fostering student creativity through independent projects, and integrating critical thinking into curricula. They also stress the significance of human oversight in reviewing LLM outputs and the necessity of a strategy focused on critical thinking and fact-checking for effective LLM integration.

Kumar et al. (Kumar et al., 2022) detail how LLM technology targets education and propose recommendations that educators may adopt to ensure academic integrity in a world with pervasive LLM tools. They emphasize the importance of fostering a genuine desire to learn and develop deep research skills among students to counteract the effects of LLM generators. They recommend to prioritize conferencing with students about their writing to foster collaboration and skill development, although this approach is challenging in large size classes.

We end this section with a description of two studies related to the impact of LLMs on higher education. Tlili et al. (Tlili et al., 2023) conducted a qualitative study to explore the impact of ChatGPT on education, structured in three phases. In the first phase, a social network analysis of tweets showed that the majority of public sentiment on social media was positive towards ChatGPT, with positive sentiments were expressed nearly twice as much as negative ones. The second phase involved interviews with 19 stakeholders who blogged about their experiences with ChatGPT. The analysis revealed that many viewed ChatGPT as transformative for education, but there were concerns about its potential to hinder innovation and critical thinking. While many found ChatGPT's responses satisfactory, some noted occasional errors and outdated information. Ethical concerns included potential plagiarism, misinformation risks, and privacy issues. The final phase presented user experiences from ten educational scenarios with ChatGPT, highlighting challenges such as cheating, truthfulness, and potential manipulation.

Sullivan et al. (Sullivan et al., 2023) analyzed 100 news articles to study ChatGPT's influence on higher education across four countries (US, UK, Australia

and NZ), published between 2020 and February 2023. Key themes included academic integrity concerns, potential AI misuse, and debates on cheating. In response, some universities reintroduced supervised exams, while others focused on assignments demanding critical thinking. Institutional policies varied, with some banning ChatGPT and others permitting its use under conditions, often citing its inevitable workplace integration. Many articles also provided strategies to combat plagiarism and promote student originality.

The numerous educational benefits of ChatGPT include personalizing learning experiences and enhancing employability as AI reshapes industries. It aids non-traditional and non-native English-speaking students, serves as a quasi-translator, and provides tools for those with disabilities, mitigating associated stigmas. However, concerns regarding ChatGPT include the potential for spreading disinformation and producing inaccurate information ('hallucinations'), as well as issues related to copyright infringement, privacy violations, and data security. While some worry about students losing critical thinking abilities, others champion the integration of AI into teaching and assignments.

Finally, the authors warn that portraying ChatGPT mainly as a cheating tool rather than a learning aid can shape public opinions about university education, influence academic reactions, and affect student perspectives on the appropriate use of this tool. Students exposed to articles about cheating with ChatGPT might be more inclined to cheat themselves. Research shows that the perception of frequent cheating opportunities increases the actual incidence of cheating among students.

Our study emphasizes achieving integrity within computer science education and outlines active strategies to enhance self-work on programming exercises. In particular, we detail practical implementations and suggestions to enhance integrity, focusing on education methods, as well as plagiarism detecting methods, used by course teams with or without the cooperation of AI developers.

3 TEXT STEGANOGRAPHY

We begin with an overview of text steganography, followed by suggestions on how it can be employed to identify AI-generated content, thereby preventing various forms of plagiarism.

Text steganography refers to methods of using text as a means to conceal information (Singh et al., 2009). In our context, we need this type of steganography, since code is represented as text, without any way to change letter sizes, fonts, etc. We can only change the characters themselves. Text steganography is notably more challenging than other forms of steganography, primarily due to the minimal redundant data available in a text file as opposed to multimedia files such as images or audio.

Numerous studies, such as (Singh et al., 2009; Por et al., 2012; Dulera et al., 2012; Roy and Manasmita, 2011; Bender et al., 1996; Agarwal, 2013; Hariri et al., 2011; Delina, 2008; Shirali-Shahreza, 2008), focus on text steganography and encryption methods. Krishnan et al. (Krishnan et al., 2017) provide a comprehensive review and classification of text steganography techniques, along with a comparison of existing approaches.

Additionally, some studies suggest using advanced algorithms and machine learning methods for the text steganography task (Xiang et al., 2020; Satir and Isik, 2012; Satir and Isik, 2014; Fang et al., 2017; Yang et al., 2020; Bhattacharyya et al., 2009). In general, the approach to text steganography relies on utilizing the unique properties of text files. These unique properties provide opportunities to hide information. For instance, one could subtly alter the text document's structure (in our case, the program code and comments) to incorporate concealed information while ensuring the changes are subtle enough not to arouse suspicion or significantly alter the output.

Additional strategies could involve the use of typographical errors, the positioning of spaces, or even the application of invisible characters. Another method could be the crafting of sentences that hold dual meanings. Here, the literal interpretation maintains the appearance of a standard document, while dedicated software will be able to reveal the information encoded in the text.

4 UTILIZING STEGANOGRAPHIC METHODS FOR FINGERPRINTED CODE

Our focus is on developing a strategy that allows ChatGPT to insert unique markers or "digital fingerprints" into its generated code snippets. Steganographic methods that manipulate white space to conceal messages (Bender et al., 1996; Por et al., 2008) appear to be suitable for this application. However, techniques like line-shift and word-shift (Roy and Manasmita, 2011; Hariri et al., 2011), which necessitate altering the spatial layout of text, are not feasible here. We will now examine multiple techniques that

ChatGPT can utilize to insert hidden messages into its output code:

- **White Spaces Encoding.** Using a combination of ”space” and ”tab” characters within blank lines can facilitate binary encoding (Por et al., 2008). This approach allows both data and control messages to be encoded by adjusting the distribution of extra spaces and tabs. For instance, one might designate ”space” to represent 0 and ”tab” to represent 1. This binary representation can convey pertinent information about the text, such as its date or the subject matter. The encoded information can then be strategically placed within the text, such as at the ends of lines or within empty lines separating code sections. Figure 3 demonstrates how a ”GPT” fingerprint can be embedded within the generated code using only invisible tabs and spaces, which can be viewed by selecting the entire code within the Colab framework. Figure 2 depicts a code generated by ChatGPT (OpenAI, 2023) that aims to produce such a tabs-and-spaces string when given input text to encrypt. Though students can detect and remove these whites pace strings, they can be dispersed throughout the code, making them hard to identify and eliminate.
- **Steganography Using Different Types of Comments.** For a language that includes two or more comment environments, for example, ”//” or ”/* */” in C++, or ”#” and extended string in Python, a coding can be performed by choosing different types of comments and using them as a signaling method, in a way determined by the teacher and the AI engine.
- **Unique Identifiers for Variables and Functions.** Various techniques can be employed to craft distinctive names for identifiers, such as variables, functions, classes, etc., enabling the detection of the code’s origin. Some examples include: using a specific and rare name, for some auxiliary variable, utilizing a mix of uppercase and lowercase letters at the start of an identifier to embed specific information; alternating between rounded or angular letters in English (Dulera et al., 2012); using identifiers with a certain common denominator, for example, with a specific sum of their ASCII values, or with a specific modulo value, when dividing by a certain k, that will be chosen in advance. Another possibility is selecting identifier names such that a hash function applied to the name yields a particular value or possesses a certain characteristic (like being even, prime, etc.).

  It is also viable to use the names of variables that have a certain common denominator, for example that the sum of the ASCII values of all the characters in the names of these variables will be 0 modulo k, for a certain k that the lecturer will choose in advance.
- **Insert Lines of Code and Comments with Certain Characteristics.** In addition to the aforementioned methods, one can encrypt a text that details the exercise’s origin and embed this encrypted version within the code comments, or implant a specific message inside the original comment, with certain specific typos (Shirali-Shahreza, 2008), etc. While it may look like a linguistic error at first glance, it would indeed represent a concealed message. Broadly speaking, various text steganography techniques – like synonyms, linguistic mistakes, or words that have different spellings in British and American English – can be integrated into the comment sections of the generated code.
- **Clear and Visible Markings.** Beyond the steganography-based solutions discussed, we can also embed clear and unequivocal markers indicating the code’s origin and purpose. This could take the form of a header comment, along with internal comments, specifying that the subsequent code was generated by the AI tool. Deleting these identifiers would demand considerable editing on the part of the student, potentially deterring plagiarism. An example of detecting work submitted with such a comment is illustrated in Figure 1.

It is recommended that the encryption be in local ranges (and not spread over the entire document), so that if the student changes something in the output program, not immediately all the encryption will be compromised. When the encryptions are local and in different places along the code, it will require thorough and even Sisyphean work on behalf of the student to identify and delete the various watermarks. Apparently, dedicated software can be created to perform this editing, and this is also an issue that the regulator will have to handle in an appropriate manner.

5 ENHANCING COOPERATION BETWEEN TEACHING STAFF AND AI COMPANIES

We proceed by describing how steganography methods can be applied and recognized by the education staff. In this study, we propose three approaches: The first involves signs that can be independently produced by an AI tool and recognized by any teacher

```python
#Code generated by ChatGPT: Visible marking
import heapq

class PriorityQueue:
    def __init__(self):
        self.queue = []
        self.index = 0

    def push(self, item, priority): #PriorityQueue push function generated by ChatGPT
        heapq.heappush(self.queue, (-priority, self.index, item))
        self.index += 1


# Testing the Priority Queue: by ChatGPT: Visible marking
pq = PriorityQueue()
pq.push("Task 1", 1)
pq.push("Task 2", 2)
pq.push("Task 3", 3)
```

Figure 1: Clear and visible marking: An illustration.

```python
def print_binary_representation(input_string):
    tabbed_line = ''
    for char in input_string:
        binary_repr = format(ord(char), '08b')  # '08b' ensures it's represe
        tabbed_repr = binary_repr.replace('0', ' ').replace('1', '\t')
        tabbed_line += tabbed_repr

    print(f"'{input_string}': {tabbed_line}")

# Test the function
input_string = "Priority Queue"
print_binary_representation(input_string)
```

Figure 2: ChatGPT generated code to provide steganography signature.

```
import heapq #encoded G
import numpy as np #encoded P
from numpy.random import rand, seed #encoded T
#Priority Queue, encoded with spaces and tabs:
class PriorityQueue:
    def __init__(self):
        self.queue = []
        self.index = 0

    def push(self, item, priority): #PriorityQueue push function generated by ChatGPT
        heapq.heappush(self.queue, (-priority, self.index, item))
        self.index += 1
```

Figure 3: White-space based steganography for a fingerprint.

or student, with or without the aid of AI detection software. The second option employs steganography applied independently by an AI tool, which then reports in a dedicated manner to a specific teacher, assuming cooperation between the teacher and the system. Lastly, the third option allows the teacher to request the AI tool to produce specific encoded messages when given certain prompts or when asked to provide a particular generated code.

How can the cooperation between an exercise providers and an AI company be carried out?

First, we assume that lecturers and teachers will be able to obtain a special license, that will enable them to cooperate on this issue with the AI company. Before giving an assignment to the students, the lecturer will inform the AI company that he is giving a certain assignment, and ask it to implant a secret message inside each code it issues at the request of someone who requests a code for this assignment, within a certain time frame, for example - within the coming week. According to the second approach, the AI company itself will report to the lecturer the message it encrypted in every assignment it received on a certain topic. If there are special licenses, the lecturer

can ask the AI company to report to him and even send him any piece of code that he issued at the request of a student on a specific topic, within a certain time frame and even from a certain geographical area. All this within the framework of the law, subsequent to the enactment of the appropriate laws in this field.

According to the third approach, the lecturer will be able to guide the AI company, on which technique to use, and what messages to implant.

The following algorithm describes a communication process for using such a method of steganography with cooperation between the teacher and the AI generator, to detect the contents it generated:

(1) The instructor creates a distinctive assignment with specific instructions that can be easily identified. For instance, the task might require printing the string "BLABLA" when a particular condition is satisfied or evaluating a unique formula using a specified number.

(2) The lecturer reports to the AI company about the exercise details and how it can be recognized. The exercise details may include some unique contents, such as unique requests (for example, printing the "BLABLA" string), the exercise date interval, and geographical area.

(3) In the second approach, the AI engine determines the coding method to employ. Meanwhile, in the third approach, the lecturer designates the specific steganographic techniques that will be implemented. At this stage, choices are made regarding the steganographic method, the text targeted for encoding, an associated encoding algorithm, a predetermined cipher for encryption, etc.

(4) When an appropriate prompt, to perform this unique task, is detected by the AI engine (according to the second or third approach), it will produce the steganographic method determined earlier within the generated code output. For example, when the AI tool is asked to print the string "BLABLA" given the above particular condition.

(5) The lecturer will be able to use dedicated software to identify code snippets that students have copied from the AI tool, using the steganographic signs implanted by the AI engine.

It is important to acknowledge that requesting the AI tool to identify users who pose a specific question could raise privacy concerns. Therefore, we propose a solution that does not necessitate such reporting. In summary, this section suggests a collaborative approach between educators and AI developers to tackle plagiarism issues exacerbated by AI tools such as ChatGPT. By leveraging the techniques of text steganography, the proposal is to embed unique "fingerprints" within the code generated by the AI tool, thus allowing educators to identify AI-generated submissions. This system can be enforced by legal frameworks, ensuring that AI developers are motivated to participate. The ultimate aim is to uphold academic integrity amidst the challenges posed by rapidly advancing technology.

6 CONCLUSIONS

In the rapidly evolving era of AI technology, we are faced with both exhilarating opportunities and daunting challenges. AI tools offer remarkable educational benefits, including interactive learning environments, personalized study modules, and extensive knowledge resources, particularly in areas like computer science and programming assistance. Yet, these advancements also bring forth significant concerns, such as the inclination of students towards academic shortcuts or an excessive reliance on automated solutions, which can impede their skill development.

In this study, we examined the complex relationship between AI tools like ChatGPT and programming education, underscoring their benefits and challenges. We particularly focused on the risk of students becoming overly dependent on these tools, potentially impeding their skill development and threatening academic integrity.

Our proposed solution involves a collaborative strategy between AI developers and educators, centered on integrating unique encrypted text markers into AI-generated responses. These markers are aimed at helping educators identify when students have used AI assistance in their programming tasks.

Our approach is designed not only to uphold academic integrity but also to foster genuine skill development among students. By clearly differentiating between student-generated work and AI-generated content, we aim to deepen student engagement with their learning materials. This method encourages a more authentic and meaningful educational experience, ensuring students truly benefit from their studies.

Future research should concentrate on developing sophisticated methods for identifying AI-generated content, ensuring its responsible application in education. This involves the creation and evaluation of algorithms for embedding watermarks in AI responses.

Testing the effectiveness of encrypted text mark-

ers in real classroom settings is a key aspect of this work, providing essential insights for the practical implementation of these methods in educational environments. Furthermore, collaboration between academia and the tech industry, supported by standardized guidelines and collaborative platforms, is vital for seamlessly integrating AI into educational systems.

It is also crucial to study the long-term effects on students who use AI tools like ChatGPT. Understanding the impact of these tools on learning outcomes, student engagement, and skill development over time is highly significant.

REFERENCES

Agarwal, M. (2013). Text steganographic approaches: a comparison. *arXiv preprint arXiv:1302.2718*.

Azaria, A., Azoulay, R., and Reches, S. (2023). Chatgpt is a remarkable tool – for experts. *arXiv preprint arXiv:2306.03102*.

Bender, W., Gruhl, D., Morimoto, N., and Lu, A. (1996). Techniques for data hiding. *IBM Systems Journal*, 35(3.4):313–336.

Bhattacharyya, D., Das, P., Bandyopadhyay, S. K., and Kim, T.-h. (2009). Text steganography: a novel approach. *International Journal of Advanced Science and Technology*, 3(1).

Cotton, D. R., Cotton, P. A., and Shipway, J. R. (2023). Chatting and cheating: Ensuring academic integrity in the era of chatgpt. *Innovations in Education and Teaching International*, pages 1–12.

Delina, B. (2008). Information hiding: A new approach in text steganography. In *Proceedings of the International Conference on Applied Computer and Applied Computational Science, World Scientific and Engineering Academy and Society (WSEAS 2008)*, pages 689–695.

Dulera, S., Jinwala, D., and Dasgupta, A. (2012). Experimenting with the novel approaches in text steganography. *arXiv preprint arXiv:1203.3644*.

Fang, T., Jaggi, M., and Argyraki, K. (2017). Generating steganographic text with lstms. *arXiv preprint arXiv:1705.10742*.

Hariri, M., Karimi, R., and Nosrati, M. (2011). An introduction to steganography methods. *World Applied Programming*, 1(3):191–195.

Kasneci, E., Seßler, K., Küchemann, S., Bannert, M., Dementieva, D., Fischer, F., Gasser, U., Groh, G., Günnemann, S., Hüllermeier, E., et al. (2023). Chatgpt for good? on opportunities and challenges of large language models for education. *Learning and Individual Differences*, 103:102274.

Khalil, M. and Er, E. (2023). Will chatgpt get you caught? rethinking of plagiarism detection.

Krishnan, R. B., Thandra, P. K., and Baba, M. S. (2017). An overview of text steganography. In *2017 Fourth International Conference on Signal Processing, Communication and Networking (ICSCN)*, pages 1–6. IEEE.

Kumar, R., Mindzak, M., Eaton, S. E., and Morrison, R. (2022). Ai & ai: Exploring the contemporary intersections of artificial intelligence and academic integrity. In *Canadian Society for the Study of Higher Education (CSSHE)*.

Lo, C. K. (2023). What is the impact of chatgpt on education? a rapid review of the literature. *Education Sciences*, 13(4).

Mijwil, M. M., Hiran, K. K., Doshi, R., Dadhich, M., Al-Mistarehi, A.-H., and Bala, I. (2023). Chatgpt and the future of academic integrity in the artificial intelligence era: a new frontier. *Al-Salam Journal for Engineering and Technology*, 2(2):116–127.

Neumann, M., Rauschenberger, M., and Schön, E.-M. (2023). "we need to talk about chatgpt": The future of ai and higher education. In *2023 IEEE/ACM 5th Int. Workshop on Software Engineering Education for the Next Generation (SEENG)*, pages 29–32.

OpenAI (2023). Chatgpt conversation. https://chat.openai.com/. Accessed: 2023-07-17.

Perkins, M. (2023). Academic integrity considerations of ai large language models in the post-pandemic era: Chatgpt and beyond. *Journal of University Teaching & Learning Practice*, 20(2):07.

Por, L. Y., Ang, T., and Delina, B. (2008). Whitesteg: a new scheme in information hiding using text steganography. *WSEAS transactions on computers*, 7(6):735–745.

Por, L. Y., Wong, K., and Chee, K. O. (2012). Unispach: A text-based data hiding method using unicode space characters. *Journal of Systems and Software*, 85(5):1075–1082.

Qureshi, B. (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*.

Roy, S. and Manasmita, M. (2011). A novel approach to format based text steganography. In *proceedings of the 2011 international conference on communication, Computing & Security*, pages 511–516.

Satir, E. and Isik, H. (2012). A compression-based text steganography method. *Journal of Systems and Software*, 85(10):2385–2394.

Satir, E. and Isik, H. (2014). A huffman compression based text steganography method. *Multimedia tools and applications*, 70:2085–2110.

Shirali-Shahreza, M. (2008). Text steganography by changing words spelling. In *2008 10th International Conference on Advanced Communication Technology*, volume 3, pages 1912–1913.

Singh, H., Singh, P. K., and Saroha, K. (2009). A survey on text based steganography. In *Proceedings of the 3rd National Conference*, pages 332–335. Bharati Vidyapeeth's Institute of Computer Applications and Management.

Sullivan, M., Kelly, A., and McLaughlan, P. (2023). Chatgpt in higher education: Considerations for academic

integrity and student learning. *Journal of Applied Learning & Teaching*, 6(1):1–10.

Susnjak, T. (2022). Chatgpt: The end of online exam integrity? *arXiv preprint arXiv:2212.09292*.

Tlili, A., Shehata, B., Adarkwah, M. A., Bozkurt, A., Hickey, D. T., Huang, R., and Agyemang, B. (2023). What if the devil is my guardian angel: Chatgpt as a case study of using chatbots in education. *Smart Learning Environments*, 10(1):15.

Xiang, L., Yang, S., Liu, Y., Li, Q., and Zhu, C. (2020). Novel linguistic steganography based on character-level text generation. *Mathematics*, 8(9):1558.

Yang, Z., Wei, N., Liu, Q., Huang, Y., and Zhang, Y. (2020). Gan-tstega: Text steganography based on generative adversarial networks. In *Digital Forensics and Watermarking: 18th International Workshop, IWDW 2019, Chengdu, China, November 2–4, 2019, Revised Selected Papers 18*, pages 18–31. Springer.