
ISSN: 2321-2152
IJMECE
International Journal of modern
electronics and communication engineering
E-Mail
editor.ijmece@gmail.com
editor@ijmece.com
www.ijmece.com

ISSN 2321-2152
www.ijmece.com
Vol 13, Issue 2, 2025

An Analysis of the Literature and Some Case Studies on the Use of Generative AI in Information and Communication Technology (ICT) Engineering Education

[1] M. Anitha, [2] SK. Salma

[1]Assistant Professor, Megha Institute of Engineering & Technology for Women, Ghatkesar.
[2] MCA Student, Megha Institute of Engineering & Technology for Women, Ghatkesar.

Abstract

To better understand how Information and Communication Technologies (ICT) engineering programs are incorporating Generative Artificial Intelligence (Gen AI) into their curricula, this article surveys the existing research on the topic. Investigated in this research is the possibility that Gen AI technologies might improve engineering education. This literature review explores the effects of Gen AI on instructional techniques, curriculum creation, and student engagement in information and communication technology engineering programs by synthesizing previous research. Based on a variety of research carried out at universities all across the globe, this article gives real-life examples of instruction, complete with student experiments and practical applications. Two important areas—programming abilities and ethics in ICT engineering education—are the primary emphasis of the illustrated scenarios, which aim to demonstrate the practical uses of Gen AI. Recognizing the significance of incorporating the study of these technologies into the curriculum, the investigation of these situations offers insightful conversation and information regarding the successful use of Gen AI in higher education. Educators, researchers, and legislators in the field of information and communication technology engineering will find this study an invaluable resource as they seek to use artificial intelligence (AI) to revolutionize engineering education.

Keywords - Generative Artificial Intelligence; Engineering education; programming; tools; ethics

INTRODUCTION

Although Generative Artificial Intelligence (Gen AI) didn't emerge until well beyond a decade ago, AI as a whole has been around for quite some time [1]. But in 2022, especially with the launch of ChatGPT, created by OpenAI (see: https://chat.openai.com/), its worldwide use and recognition surged dramatically. Particularly in the realm of education, Gen AI technologies have shown to be quite beneficial. These tools have a wide range of potential applications in education, including the creation of complex and realistic content, the transformation of different types of data, the detection of students' emotional states, and the development of adaptive assessments that change based on students' performance [2]. Taking into account four distinct groups, Chiu et al. [3] determine the primary function of AI in the field of education: (i) Education: (a) facilitating assignments according to students' abilities; (b) facilitating interactions between humans and machines; (c) evaluating comments made by students on their work; and (d) fostering flexibility and engagement in virtual spaces. (ii) In the realm of instruction, we aim to (a) improve management platform performance, (b) provide students with convenient and personalized services, and (c) support educational decision-making with evidence when it comes to adaptive teaching strategies, teacher professional development, and student performance. In the realm of assessment, we provide automatic marking and student performance. There are mixed feelings on the use of Gen AI in the classroom, including apprehension about potential ethical issues and test cheating, and a general openness to new ideas. Educators in higher education have differing views on the topic, but a growing consensus is that these emerging technologies must be taught in the classroom if they are to spur further innovation [4].

With an emphasis on bolstering students' programming abilities and addressing ethical concerns surrounding these technologies, this article mainly discusses how Gen AI may be included into ICT engineering curricula. Our thorough literature analysis on the topic of Gen AI in ICT engineering education led us to a key observation, which we want to shed light on. It is now clear that the most prevalent use of Gen AI in this educational environment is to enhance programming abilities. While not limited to ICT engineering, the highlighted examples of larger educational situations for Gen AI are important and might be useful across multiple disciplines. Some examples include using Gen AI as an expert in the field for interdisciplinary inquiry-based learning [5], checking AI-generated writings for mistakes [6], and investigating the relationship between academic honesty and AI ethics [7]. In the field of information and communication technology engineering, there are a handful of instances of how these tools have been used to teach machine learning ideas [9] and to facilitate creative processes and digital prototyping [8]. We have decided to center our discussion on Gen AI's role in improving software engineering activities because of its major attention and potential influence in this area. Additionally, we address the ethical issues of AI in engineering educational contexts as a result of its effects on instruction and student learning. Using this method, we may explore the real-world uses and consequences of Gen AI in promoting academic honesty and advanced programming skills in the field of information and communication technology engineering.

LEVERAGING GEN AI TO ENHANCE SOFTWARE ENGINEERING SUPPORT

A key function of Gen AI is code translation into other programming languages, code generation, and recommendation offering. Because of this, efficiency in code generation is sped up and productivity is increased. It is considered that using Gen AI to do simple tasks would not help students acquire essential programming abilities, which is a major issue for educators [10]. Users unfamiliar with programming may actually benefit from these AI-based coding tools, since they may be used for self-directed learning [11]. To familiarize students with new program implementations and functions, and to instruct them in their proper use and syntax. As part of a larger effort to move away from teaching students basic "hello world" programming and toward a more in-depth curriculum, Ozkaya [12] argues that aspiring software engineers need training in several areas, such as how to read and understand large amounts of code, how to find and fix problems in unfamiliar codebases, and how software works as a whole. Emphasizing the use of Large Language Models (LLMs) and comparable Gen AI-driven tools in teaching ICT engineering students when to trust, how to produce trustworthy evidence, how to efficiently and accurately measure trust, and how to improve AI assistants should be a top priority. Students should also understand how to handle data similarly to code and how to incorporate these components into growing systems. For example, research on Chat GPT's efficacy on typical programming tasks has shown promising results; nevertheless, its attention span is one area where it has shown certain limits. Chat GPT was unable to apply its extensive knowledge to the real issue because of the too specific and lengthy explanations that limited its attention and prevented it from generalizing to new and undiscovered situations [13]. This knowledge would allow for the development of programming skill tests that are intentionally difficult for AIs to complete. This safeguard would deter students from depending on AIs and instead force them to work together to complete the tasks or use them as a tool for another set of exercises, encouraging the development of diverse skill sets.

GENERATIVE AI TOOLS TO SUPPORT PROGRAMMING TASKS

Students increasingly rely on Chat GPT, making it a top Gen AI tool. A set of inquiries has been made in order to comprehend the operation and reactions of this technology, specifically ChatGPT. With the goal of learning more about how the system works and investigating the benefits and drawbacks of using Gen AI in the classroom [14]. There have been many releases of ChatGPT so far. Even though it trained on a massive text and code dataset, the first one, ChatGPT3.5, isn't always the fastest, particularly when creating large volumes of text. The most current and widely used version, ChatGPT4, is built on the GPT-4 language model, which is the most powerful language model ever made, outperforming all previous versions in terms of speed and accuracy. Unfortunately, at the moment, it's only accessible to people who pay for it. The purpose of this section is to provide a concise overview of certain Gen AI and AI technologies that have been found to be useful in helping various software engineering jobs (refer to Table 1).

TABLE I. Tools SELECTION OFARTIFICIAL INTELLIGENCE TOOLS FOR PROGRAMMING

IJMECE
International Journal of modern electronics and communication engineering
ISSN 2321-2152
www.ijmece.com
Vol 13, Issue 2, 2025

Tools	Function
Amazon CodeWhisperer	Enabling a faster and more secure creation of applications. Providing real-time code hints, reference tracking and security analysis
Android Studio Bot	Generates code, fix bugs, and provide answers to questions related to Android development. It is an essential tool for streamlining the development process and getting fast, accurate support.
AskCodi	It generates code, answers programming questions and provides useful code hints.
AI Helper Grip	AI-powered wizard that facilitates the generation of SQL queries for users with different skill levels. It also allows users to save and share generated SQL queries for later use.
Bugasura	Application bug reporting tool that allows developers to detect and fix bugs at high speed.
Chat Gpt	Generates natural language answers to questions and queries. Specifically for programmers, it is useful in providing key information and answers on programming and other computer-related topics, which can save them time and help in solving technical problems.
Codiga	It allows the creation, use and sharing of smart code snippets, performing analysis to detect and correct errors, generating error reports and allowing the creation of customized code analysis rules.
CodeClimate	Code analysis tool that uses AI for programmers to identify code quality issues, security vulnerabilities and potential bugs. It also offers an integration with GitHub to provide automatic feedback on code quality.
CodeGuru	Amazon Web Services artificial intelligence tool that uses machine learning to improve code quality and performance. It provides suggestions for optimizing code, detecting performance issues and improving security.
CodeWP	Quickly generate valid code for a variety of tasks. Produces efficient and secure code that can be edited as needed.
DeepCode	Code analysis tool that uses artificial intelligence to find bugs and security vulnerabilities in code.
Deep Learning Studio	Offers tools for neural networking, image processing and natural language processing.
GitHub Copilot	Completion tool that assists developers by suggesting contextually relevant code snippets as they write.
OpenAI Codex	Code generator based on human-like text, being able to translate code into various programming languages.
Replit	Provides an interactive space for users to code, collaborate and learn collectively. Accelerates coding with advanced online hints, enabling coding learning directly in the document.
Sourcegraph Cody	It can generate code snippets or complete functions based on given instructions, explain complex code structures or functions in simple terms, and analyze code blocks for bugs and potential problems.

Tools	Function
SinCode	Script generation assistant but is also capable of generating AI code, as well as performing an error search.
Tabnine	Allows writing code quickly and efficiently improving productivity through intelligent code suggestions, line completion and code functions. It also completes code in natural language improving the quality and consistency of the suggested code, aligning with the user's coding patterns.
WPCode	A WordPress plugin for code snippets. Facilitates and protects the addition of custom WordPress functions through code snippets, allowing you to reduce the number of plugins on your site.

Application of Gen Ai to Support Programming Skills in Software Engineering

So far, research on ICT engineering education has been mixed, with some publications reporting experiments and others discussing practical applications. In order to help in software engineering and programming, these sections provide a summary of several case studies: Here, 12 activity sheets were developed using ChatGPT as an automated question generating (AQG) tool for Java programming classes at the University of Education Ludwigsburg, which caters to beginners to intermediates. Students were asked to solve the tasks and assess their own drafting skills in order to evaluate its success. Even though they didn't seem to have been authored by a Gen AI, the produced tasks were helpful in taking into account the level required in a university setting. Creating exercises that practice identifying errors and mapping them to their classes in given code was not so good, but ChatGPT was especially fast at creating good exercises about control flow structures, understanding and using given APIs, and inheritance [15]. Students at Munich's Ludwig-Maximilians-Universität have shown, via the submission of real-world projects, how well the language model GPT-3.5 provides individualized feedback for programming tasks, such as recommendations for style and code correction [16]. With a success rate of 73% in differentiating between right and wrong submissions and good comments in 47% of instances, it clearly excels at spotting wrong submissions. Even while it could catch functional and syntactic mistakes, it did notice some inaccuracies and did not always follow the assignment guidelines. Use GPT-3.5 with care when implementing completely automated student feedback in programming projects because of dependability problems. Nevertheless, ChatGPT4, in its present form, has great potential as an invaluable resource for TAs to pre-assess extensive assignments, allowing for rapid mistake detection and the generation of drafts for individualized comments. Here we have another case study involving two experiments designed to assess performance on different coding tasks. These experiments were carried out by a professor from Stanford University's Department of Psychology in collaboration with an assistant professor and a student researcher from the University of California, Berkeley's Department of Linguistics.

ISSN 2321-2152
www.ijmece.com
Vol 13, Issue 2, 2025

One experiment demonstrated GPT-4's value for researchers lacking in quick technical expertise by testing its capacity to produce executable code for data science tasks. To show that GPT 4 can produce cleaner, more standards-compliant code, the second experiment looked at its capacity to restructure without losing its original meaning. It was discovered, however, that GPT-4's performance may be much better when combined with other formatting tools. The last experiment examined GPT-4's test generation capabilities; it found that while it could create testing code with acceptable coverage, test failure was common, necessitating further debugging. In spite of the great outcomes, human oversight is still necessary to verify the final code is valid and accurate, even with state-of-the-art AI systems like GPT-4 [17]. So far, we have mostly focused on ChatGPT, but other generational AIs like Copilot (see Table 1) have also been investigated in higher education information and communication technology engineering contexts. Two institutions, one in New Zealand and the other in India, have used a dataset of 166 programming tasks to evaluate Copilot's efficacy [18]. Copilot expertly resolves almost half of these issues on its first try, according to the evaluation. In addition, by modifying the problem description using natural language, it gets a success rate of 60% for the remaining challenges. Students and instructors may check whether the original input produces the intended result by copying and pasting textual problem statements provided as part of the assessment process. When this doesn't work, users may try other inputs until they succeed in getting the desired code. An experiment was conducted by the Department of Computer Science at Prince Sultan University in Saudi Arabia. The students in the control group had access to programming course materials without Internet, whereas the experimental group had access to ChatGPT. Both teams had a limited amount of time to complete a series of programming tasks. Using ChatGPT gave students a score boost, but their submissions were flawed due to inconsistencies and errors, which affected their total performance [19]. Incorporating AI into higher education might provide both benefits and problems, especially when considering the significance of students' work in sustaining ethical standards.

Consistent with these results, an investigation of Codex revealed that it performed similarly to the top quartile of pupils when it came to solving Computer Science 2 (CS2) problems. With fewer edge situations and no need to adjust existing code, Codex performed very well on queries that were well defined and eloquently phrased. This proves that Codex can handle the difficult challenges that come up in Computer Science 1 (CS1) classes. There is still considerable debate, nonetheless, about the cutoff point for question complexity to have a substantial impact on Codex performance in the field of computer science education [20]. The emphasis has shifted from code writing challenges to the resolution of prompt problems, which is a new tool used in computer science courses to teach students how to create successful prompts for artificial intelligence code generators. 1340 This license is only valid for usage at Zhejiang University. This document was retrieved from IEEE Xplore on December 19, 2024 at 05:48:08 UTC. Conditions are applicable. In order to read, understand, and evaluate code that is created by LLM [21]. It has also been said that Gen AIs may make jobs longer rather than shorter by helping with coding and offering more general ways to get the job done. The production of unwanted outputs makes it more difficult to achieve the intended outcome, which is the reason for this. Therefore, the job has to be rewritten many times. Thus, it is critical to communicate creatively with the system and guarantee correct AI replies by speaking the same "language" [22]. The goal of this kind of assignment, called a "Prompt Problem," is to have students provide natural language prompts that an LLM may use to generate the right code to solve a given issue. University of Auckland (New Zealand) students are taught these skills using an exercise that shows them an input and its matching output. Then, they are asked to use ChatGPT to develop a program that can convert the input to the output. In order to discourage participants from just copying and pasting the issue description, the visual representation is designed to encourage them to come up with an appropriate prompt on their own [23]. Problems occurred throughout the exercise for a number of reasons, the most common of which were participants' approaches to prompting. Starting with a hello but offering no details about the code they wanted the LLM to produce, other students started discussions with incomplete instructions. Here, instead of replicating the original request, students in an introductory Python programming course tried iteratively to progressively describe what they meant. The fact that some participants had trouble understanding the situation meant that their prompts were ill-conceived and had little likelihood of producing the expected effect. So, a lot of the time, while interacting with ChatGPT, people would ask for improvements that wouldn't work. Additionally, it was observed that users would often ask ChatGPT to solve an issue, check its accuracy using an automated assessment tool, and then copy and paste the failed test cases back into ChatGPT word for word, without providing

IJMECE International Journal of modern electronics and communication engineering
ISSN 2321-2152
www.ijmece.com
Vol 13, Issue 2, 2025

any extra context or hints. Thirdly, when the task was given, some pupils had a total misunderstanding of it [23].

GENERATIVE AIANDETHICS IN ICTENGINEERING EDUCATION

The need of teaching young engineers ethical standards is growing as Gen AI changes the face of technical progress. To do so, one must be well-versed in the technical aspects of Gen AI as well as the ethical considerations that surround its creation and use. An engineering education perspective on Gen AI ethics aims to train engineers who are knowledgeable about AI technology and its applications, but who are also sensitive to the ethical challenges and opportunities presented by designing intelligent systems that promote human values and the common good, taking into account concerns like transparency and bias as well as the societal effects of AI-driven technologies. One issue to think about is that some languages are underrepresented, including indigenous languages. This is because big language models get their training data from internet datasets that have a lot of English and very little indigenous material. Banning this technology in the sake of reducing language inequality would, however, cause a new kind of inequality to arise. Regarding this matter, and to provide a general example that may be used to several fields, a research [23] conducted at Stanford University (USA) sought to evaluate the effectiveness of numerous popular GPT detectors by analyzing literary examples from both native and non-native English speakers. According to the results, these detectors reliably mistake samples of non-native English writing for AI-generated content, whereas native writing examples are correctly recognized. Additionally, it was shown that basic prompting techniques might both reduce this bias and successfully evade GPT detectors, implying that GPT detectors could inadvertently criticize authors who use limited language expressions.

The findings highlight the need for further discussion on the moral consequences of using ChatGPT content detectors in educational or assessment contexts, especially when they have the potential to unintentionally discriminate against or otherwise marginalize non-native English speakers in global conversations. How the training data is structured greatly affects the level of bias in AI. Huge, annotated datasets are essential for many machine-learning jobs. Annotation techniques, however, could unintentionally add cultural, ethnic, and gender biases. A lack of geodiversity occurs when training data is mostly derived from one country. Maximizing the overall prediction accuracy for the training data is the goal of any machine learning program's design. In order to improve overall accuracy, the algorithm adjusts its optimizations based on whether specific groups are overrepresented in the training data. Computer scientists often use 'test' datasets to assess algorithms, but these are often just randomly selected portions of the training set, which means they might still be biased. Hence, building training datasets requires a combination of technical rigor and social consciousness. To prevent underrepresentation of some groups, it is essential to ensure variety in these databases. To do this, we must go beyond binary categories that ignore the complexity of gender and race, such as "man" and "woman" [24]. Our research has led us to the following concrete instances that demonstrate the incorporation of ethical concerns within ICT engineering curricula: A study carried out by researchers from Northeastern University and the University of Colorado at Boulder poses the vital issue of why, despite the relevance of social and ethical considerations in computer system design, these topics were often neglected in computer science curricula. Programs used different methods since there weren't clear standards on how to include these issues into the curriculum. One usual approach was to provide students with electives in ethics or professionalism during their junior or senior years. Limitations for non-majors, the possibility of separating ethics from technical practice, and the postponement of the introduction of ethical issues in a student's educational path were some of the possible problems highlighted by the research as being associated with depending just on standalone ethics lectures. In response to these concerns, a pilot program was launched to include ethics into first-year computer science courses. We stressed the significance of ethics from the outset of their computer science education in this method, with the goal of reaching a larger student population, including those who may not pursue additional computing courses. Ethical challenges and ideas were used to contextualize the current tasks in the pilot program. Instructor and TA observations, as well as a post-class poll of students, formed the basis for a discussion of the intervention's pros and cons [25]. Notable educational institutions are currently offering new ethics courses. A new course is being given jointly by Harvard University and the Massachusetts Institute of Technology on the ethics and regulation of artificial intelligence. "Ethical Foundations of Computer Science" is a newly offered course at the University of Texas at Austin with the goal of requiring it of all computer science majors in the future. At Stanford University, three faculty members and a research fellow are working on a computer science ethics course right now. This is all in an effort to make sure that the next generation of techies and politicians think about the consequences of technologies like self-driving vehicles and autonomous weaponry before they hit the market [26].

CONCLUSIONS

Curriculums need to be adjusted to include unique skills and AI literacy in order for AI tools to be effectively integrated into ICT engineering programs in higher education. Teachers and students both need to be able to recognize the biases and limits of AI systems and know how to evaluate the credibility of AI-generated material. Students majoring in engineering, in particular, should work on developing their critical thinking abilities so that they can ask pertinent questions and work successfully with AI technologies. One obstacle to developing a standardized method is the lack of an existing, widely recognized paradigm for AI literacy, as mentioned in [27]. Both the potential benefits and drawbacks of Gen AI must be fully understood in order to make the most of this technology. When certain AI products need costly memberships or access fees, it might lead to unequal access because only those with the means can afford them. The digital gap is growing as a result of this disparity, which in turn keeps educational inequality at bay. The fact that certain countries have limits or prohibitions on some (Gen) AI technology makes matters worse, since learners in such areas don't have access to the advantages that these technologies may provide. Artificial intelligence technologies can only learn from data that is both high-quality and diverse, hence their dependence on external data sources is a major drawback. Unfair or incorrect results may result from the reproduction and amplification of biases in the training data. The study's chosen application cases and trials in real-world educational settings (refer to Table 2) assist to highlight the technologies' strengths and weaknesses, allowing for their improvement and better adaptation to educational settings.

TABLE II. Categories OF REVIEWED PAPERS

Categories	References
History of Gen AI	[1] Van der Zant, T., Kouw, M., & Schomaker, L. (2013)
Impact of Integrating AI into the curriculum	[3] Chiu, T. K., Xia, Q., Zhou, X., Chai, C. S., & Cheng, M. (2023) [4] Dai, Y., Liu, A., & Lim, C. P. (2023) [27] Lee, S., de Waard, I. (2023)
Gen AI use in various educational disciplines	[2] Baidoo-Anu, D., & Owusu Ansah, L. (2023) [5] Auerbach, J. E., Concordel, A., Kornatowski, P. M., & Floreano, D. (2018). [6] Alharbi, W. (2023) [7] Hernández-Leo, D. (2023)
Analysis of AI usage in ICT engineering education	[8] Bilgram, V., & Laarmann, F. (2023) [9] Lu, W. Y., & Fan, S. C. (2023) [10] Lau, S., & Guo, P. (2023) [11] Jung, H. W. (2020) [12] Ozkaya, I. (2023) [13] Tian, H., Lu, W., Li, T. O., Tang, X., Cheung, S. C., Klein, J., & Bissyandé, T. F. (2023) [14] Qadir, J. (2023) [15] Speth, S., Meißner, N., & Becker, S. (2023) [16] Azaiz, I., Deckarm, O., & Strickroth, S. (2023) [17] Poldrack, R. A., Lu, T., & Beguš, G. (2023) [18] Denny, P., Kumar, V., & Giacaman, N. (2023) [19] Qureshi, B. (2023). [20] Finnie-Ansley, J., Denny, P., Luxton-Reilly, A., Santos, E. A., Prather, J., & Becker, B. A. (2023). [21] Denny, P., Leinonen, J., Prather, J., Luxton-Reilly, A., Amarouche, T., Becker, B. A., & Reeves, B. N. (2023) [22] Jonsson, M., & Tholander, J. (2022). [23] Denny, P., Leinonen, J., Prather, J., Luxton-Reilly, A., Amarouche, T., Becker, B. A., & Reeves, B. N. (2023 [23] Liang, W., Yuksekgonul, M., Mao, Y., Wu, E., & Zou, J. (2023)
Ethics in AI education	[24] Zou, J., & Schiebinger, L. (2018) [25] Garrett, N., Beard, N., & Fiesler, C. (2020) [26] Singer, N. (2018)

REFERENCES

[1]. T. van der Zant, M. Kouw, and L. Schomaker, "Generative artificial intelligence," in Studies in Applied Philosophy, Epistemology and Rational Ethics, Springer Berlin Heidelberg, 2013, pp. 107-120.

[2]. D. Baidoo-Anu and L. Owusu Ansah, “Education in the era of Generative Artificial Intelligence (AI): Understanding the potential benefits of CHATGPT in promoting teaching and learning,” in SSRN Electronic Journal, 2023.

[3]. T. K. F. Chiu, Q. Xia, X. Zhou, C. S. Chai, and M. Cheng, “Systematic literature review on opportunities, challenges, and future research recommendations of Artificial Intelligence in Education,” Computers and Education: Artificial Intelligence, vol. 4, 2023, p. 100118.

[4]. Y. Dai, A. Liu, and C. P. Lim, "Reconceptualizing Chatgpt and Generative AI as a student-driven innovation in Higher Education," Procedia CIRP, vol. 119, 2023, pp. 84–90.
[5]. J. E. Auerbach, A. Concordel, P. M. Kornatowski, and D. Floreano, "Inquiry-Based Learning with RoboGen: An open-source software and hardware platform for Robotics and artificial intelligence," IEEE Transactions on Learning Technologies, vol. 12, no. 3, Jul. 2019, pp. 356–369.
[6]. W. Alharbi, "Ai in the Foreign Language Classroom: A pedagogical overview of Automated Writing Assistance Tools," Education Research International, vol. 2023, Feb. 2023, pp. 1–15.
[7]. D. Hernández-Leo, "ChatGPT and Generative AI in Higher Education: User-Centered Perspectives and Implications for Learning Analytics," 2023.
[8]. V. Bilgram and F. Laarmann, "Accelerating innovation with Generative AI: Ai-Augmented Digital Prototyping and innovation methods," IEEE Engineering Management Review, vol. 51, no. 2, Jun. 2023, pp. 18–25.
[9]. W.-Y. Lu and S.-C. Fan, "Developing a weather prediction project based machine learning course in facilitating AI learning among high school students," Computers and Education: Artificial Intelligence, vol. 5, 2023, p. 100154.
[10]. S. Lau and P. 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," in Proceedings of the 2023 ACM Conference on International Computing Education Research, vol. 1, 2023, pp. 106-121.
[11]. H.-W. Jung, "A Study on the Current State of Artificial Intelligence Based Coding Technologies and the Direction of Future Coding Education," International Journal of Advanced Culture Technology, vol. 8, no. 3, Sep. 2020, pp. 186–191.
[12]. I. Ozkaya, "Application of large language models to software engineering tasks: Opportunities, risks, and implications," IEEE Software, vol. 40, no. 3, May 2023, pp. 4–8.
[13]. T. Tian, W. Lu, T. O. Li, X. Tang, S. C. Cheung, J. Klein, and T. F. Bissyandé, "Is ChatGPT the Ultimate Programming Assistant--How far is it?," arXiv preprint arXiv:2304.11938, 2023.
[14]. J. Qadir, "Engineering education in the era of chatgpt: Promise and pitfalls of Generative AI for Education," 2023 IEEE Global Engineering Education Conference (EDUCON), May 2023 pp. 1-9.
[15]. S. Speth, N. Meißner, and S. Becker, "Investigating the Use of AI Generated Exercises for Beginner and Intermediate Programming Courses: A ChatGPT Case Study," in 2023 IEEE 35th International Conference on Software Engineering Education and Training (CSEE&T), Aug 2023, pp. 142-146.
[16]. I. Azaiz, O. Deckarm, and S. Strickroth, "Ai-Enhanced Auto correction of programming exercises: How effective is GPT-3.5?" International Journal of Engineering Pedagogy (iJEP), vol. 13, no. 8, Dec. 2023, pp. 67–83.
[17]. R. A. Poldrack, T. Lu, and G. Beguš, "AI-assisted coding: Experiments with GPT-4," arXiv preprint arXiv:2304.13187, 2023.
[18]. P. Denny, V. Kumar, and N. Giacaman, "Conversing with copilot: Exploring prompt engineering for solving CS1 problems using natural language," Proceedings of the 54th ACM Technical Symposium on Computer Science Education vol. 1, Mar. 2023.
[19]. B. Qureshi, "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.
[20]. J. Finnie-Ansley et al., "My AI Wants to Know if This Will Be on the Exam: Testing OpenAI's Codex on CS2 Programming Exercises," in Proceedings of the 25th Australasian Computing Education Conference, Jan. 2023, pp. 97-104.
[21]. P. Denny et al., "Prompt Problems: A New Programming Exercise for the Generative AI Era," arXiv preprint arXiv: 2311.05943, 2023.
[22]. M. Jonsson and J. Tholander, "Cracking the code: Co-coding with AI in creative programming education," in Proceedings of the 14th Conference on Creativity and Cognition, Jun. 2022, pp. 5-14.
[23]. W. Liang, M. Yuksekgonul, Y. Mao, E. Wu, and J. Zou, "GPT detectors are biased against non-native English writers," Patterns, vol. 4, no. 7, p. 100779, Jul. 2023.
[24]. J. Zou and L. Schiebinger, "AI can be sexist and racist — it's time to make it Fair," Nature, vol. 559, no. 7714, pp. 324–326, Jul. 2018.
[25]. N. Garrett, N. Beard, and C. Fiesler, "More than 'If Time Allows' the Role of Ethics in AI Education," in Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society, Feb. 2020, pp. 272-278.
[26]. N. Singer, "Tech's ethical 'dark side': Harvard, Stanford and others want to address it," NY Times, 2018.
[27]. S. Lee and I. de Waard, "AI toolkit for Educators," EIT InnoEnergy Master School Teachers Conference 2023, 2023.