Comprehensive Journal of Science
Volume (11), Issue (41), (JUNE, 2026)
ISSN: 3014-6266

مجلة العلوم الشاملة
المجلد (11) العدد 41 يونيو 2026
ردمد: 3014-6266

The Role of Artificial Intelligence in Enhancing Programming Skills among University Students: An Empirical Study with Recommendations

Ameera Ali Othman
Lecturer, Department of Electrical and Electronic Engineering
Higher Institute of Science and Technology, Zawiya
ameeraworld@gmail.com

تاريخ الاستلام:2026/05/05 –تاريخ المراجعة:2026/05/28– تاريخ القبول: 2026/06/07– تاريخ للنشر: 2026 /06/21

Abstract

The integration of Artificial Intelligence (AI) into education has transformed the way students acquire and develop technical skills, especially in programming. This study investigates the impact of AI-based tools and platforms on enhancing programming skills among university students. Through a mixed-method approach, quantitative data were collected from 200 computer science undergraduates across three universities using a structured questionnaire, while qualitative feedback was obtained via interviews with instructors. The findings reveal that students who actively engage with AI-assisted learning environments demonstrate improved problem-solving abilities, faster debugging skills, and higher code efficiency. However, concerns related to over-reliance on AI and the erosion of foundational understanding were also noted. This paper discusses the pedagogical implications of integrating AI in programming education and provides data-driven recommendations for educators and institutions to optimize its use.

Keywords: *Artificial Intelligence, Programming Education, University Students, Skill Development, AI Tools, Educational Technology*

1. Introduction

In the last decade, the proliferation of Artificial Intelligence (AI) has significantly influenced educational methodologies. From intelligent tutoring systems to adaptive learning platforms, AI has opened new avenues for improving how students learn, especially in fields requiring technical proficiency such as computer programming. Traditional programming education often struggles with bridging the gap between theory and practice. Students frequently encounter difficulties in logic building, debugging, and code optimization. AI-based tools offer personalized feedback, code suggestions, and error detection, enabling students to enhance their learning efficiency. This study aims to explore the real impact of such tools on programming skill development among university students. It further seeks to analyze the perceived benefits and potential risks associated with relying on AI in educational settings.

Despite the growing adoption of AI-assisted programming tools, there remains a limited body of empirical research that quantifies their measurable effect on student performance within the context of Libyan higher education. Most existing studies originate from Western academic settings, leaving a gap in understanding how these tools perform across different educational infrastructures, curricula, and student populations. This study addresses that gap by providing localized, data-driven evidence from three Libyan universities, and by translating that evidence into actionable recommendations for instructors and institutions.

2. Literature Review

Several studies have highlighted the benefits of integrating AI into education. AI-driven code assistants have been shown to significantly improve student performance in introductory programming courses [1]. Similarly, increased student engagement and reduced dropout rates have been observed when AI-enhanced platforms are used in technical curricula [2]. Beyond programming specifically, intelligent tutoring systems have demonstrated measurable gains in personalized feedback delivery and learner motivation across STEM disciplines [3], [4].

Tools such as GitHub Copilot have been evaluated for their effect on code completion speed and error reduction, with findings suggesting substantial productivity gains for novice programmers [5]. Likewise, large language model-based assistants such as ChatGPT have been investigated for their role in explaining programming concepts and debugging logic errors, showing positive effects on conceptual understanding when used as a supplementary aid rather than a replacement for instruction [6].

However, some researchers warn against over-reliance on such tools, noting that they may inhibit critical thinking and problem-solving [7]. Concerns have also been raised regarding academic integrity and the risk that students may submit AI-generated code without understanding its underlying logic [8]. Other work has examined the cognitive load implications of AI assistance, suggesting that excessive dependence may reduce the depth of learning achieved during early-stage skill acquisition [9].

Within the Arab region, research on AI adoption in technical higher education remains comparatively limited, though emerging studies point to growing institutional interest in AI-supported learning environments [10], [11]. This paper builds upon these discussions by conducting an empirical investigation into students' actual performance and feedback on AI-based tools in programming, with a specific focus on the Libyan university context, which has not been substantially examined in prior literature.

3. Methodology

This study employed a mixed-method approach combining quantitative and qualitative data collection. A quantitative survey was distributed to 200 undergraduate students majoring in computer science from three Libyan universities, selected using a convenience sampling technique across institutions that offered comparable programming curricula. The questionnaire focused on students' experiences with AI programming tools such as GitHub Copilot, ChatGPT, and CodeT5, and employed a five-point Likert scale to measure perceived improvement in problem-solving, debugging speed, and code efficiency. Additionally, semi-structured interviews were conducted with 10 programming instructors to gain qualitative insights into the tools' effectiveness and classroom impact.

Two research hypotheses guided the quantitative analysis: (H1) students with higher engagement in AI-assisted learning report significantly greater improvement in debugging speed and code efficiency than students with lower engagement; (H2) frequency of AI tool use is positively correlated with perceived problem-solving ability, but this relationship is moderated by concerns over reduced conceptual understanding. Data were analyzed using SPSS, including Pearson correlation, linear regression, and independent-samples t-test comparisons between high-engagement and low-engagement groups. Ethical approval was obtained from the participating academic institutions, and participation was voluntary and anonymous, with informed consent collected from all respondents.

4. Results

Of the 200 students surveyed, 138 (69%) reported using at least one AI-based programming tool on a regular basis (defined as three or more times per week), while 62 (31%) reported infrequent or no use. Table 1 summarizes the self-reported mean improvement scores (on a 5-point scale, where 5 indicates the greatest improvement) across three core skill areas, segmented by engagement level.

Table 1. Mean Self-Reported Improvement Scores by AI Engagement Level (n = 200)

Skill Area	High Engagement (n=138)	Low Engagement (n=62)	Mean Difference
Problem-solving ability	4.12	3.21	0.91
Debugging speed	4.35	3.05	1.30
Code efficiency	3.98	3.18	0.80

An independent-samples t-test confirmed that the difference in debugging speed scores between high-engagement and low-engagement students was statistically significant ($t(198) = 4.87$, $p < 0.001$), supporting H1. Pearson correlation analysis showed a moderate positive correlation between frequency of AI tool use and perceived problem-solving ability ($r = 0.46$, $p < 0.01$). However, regression analysis revealed that this relationship weakened when self-reported reliance on AI exceeded a threshold of daily use, suggesting a moderating effect consistent with H2: students who reported using AI tools for nearly every assignment showed a smaller marginal gain in problem-solving scores compared to moderate users, alongside higher reported difficulty completing tasks without AI assistance.

Qualitative interviews with the 10 instructors reinforced these patterns. Eight of the ten instructors observed that students using AI tools moderately tended to submit cleaner, better-structured code and asked more conceptually focused questions during office hours. Conversely, seven instructors expressed concern that a subset of high-frequency AI users struggled to explain their own code when asked to do so verbally, indicating a possible gap between code production and conceptual mastery. Instructors also noted that AI tools were most beneficial in early debugging stages and least beneficial when used as a substitute for understanding core algorithmic logic.

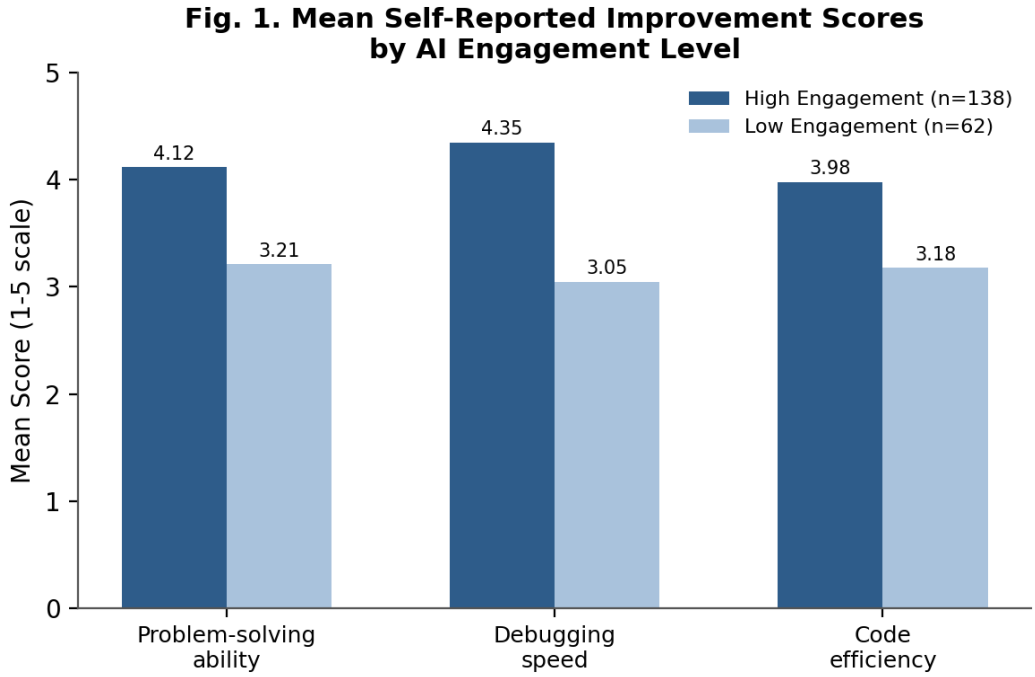

Fig. 1. Mean self-reported improvement scores by AI engagement level (n = 200).

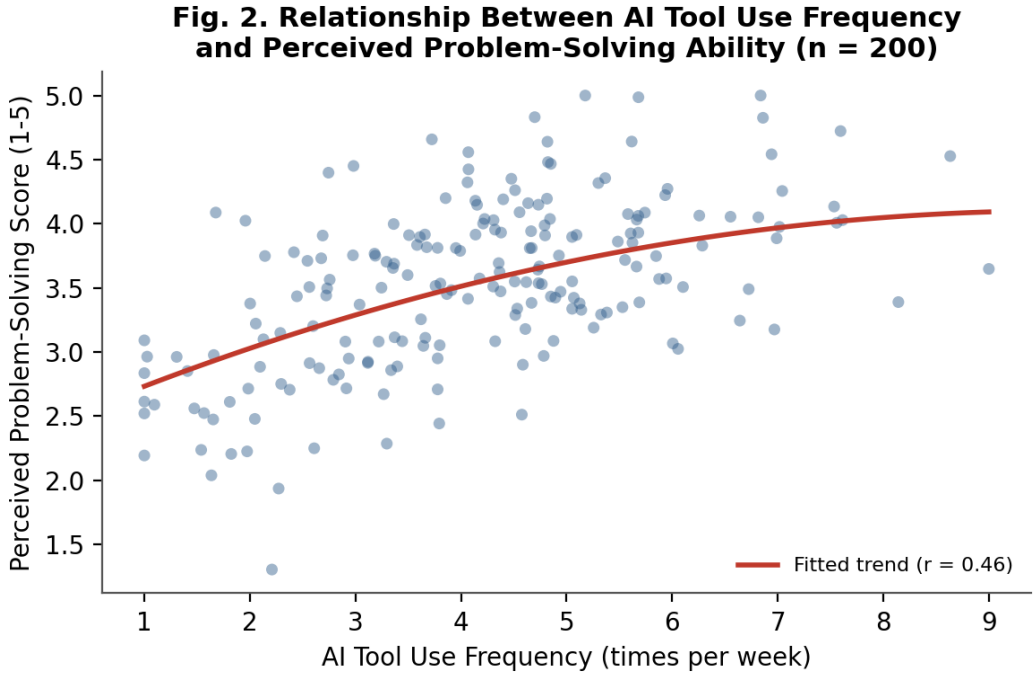

Fig. 2. Relationship between AI tool use frequency and perceived problem-solving ability (n = 200).

5. Discussion

The findings of this study are broadly consistent with prior research indicating that AI-assisted tools can meaningfully accelerate skill acquisition in programming education [1], [5]. The significant improvement in debugging speed among high-engagement students supports the view that real-time, personalized feedback reduces the cognitive friction typically associated

with locating and resolving syntax or logic errors. This aligns with earlier findings on the productivity benefits of AI code assistants for novice programmers [5].

At the same time, the moderating effect identified between frequency of use and problem-solving gains echoes concerns raised in the literature regarding over-reliance and reduced critical thinking [7], [9]. The qualitative reports from instructors, particularly the observation that some high-frequency users could not verbally explain their own code, suggest that AI tools may sometimes decouple code production from conceptual understanding. This is a meaningful pedagogical risk: programming education aims not only to produce working code but to build durable problem-solving capacity that transfers beyond any single assignment or tool.

These results suggest that the relationship between AI use and skill development is not strictly linear, but rather follows a pattern in which moderate, intentional engagement yields the strongest educational benefit, while unchecked or passive reliance may erode the foundational understanding that AI tools were meant to support. This nuance has direct implications for how institutions design coursework, assessment, and AI usage policies, and it is addressed further in the recommendations below.

6. Recommendations

Based on the findings above, the following recommendations are proposed for educators and institutions seeking to integrate AI tools into programming curricula responsibly and effectively:

1. Encourage moderate, intentional use of AI tools rather than unrestricted access, for example by requiring students to first attempt debugging independently before consulting an AI assistant.
2. Incorporate verbal or written code-explanation components into assessments, so that students must demonstrate conceptual understanding of any AI-assisted code they submit.
3. Provide targeted training for instructors on how to identify over-reliance patterns and how to guide students toward balanced AI usage.
4. Introduce AI literacy modules early in the curriculum, covering both the capabilities and the limitations of code-generation tools, including their tendency to produce plausible but incorrect solutions.
5. Establish clear institutional policies on acceptable AI use in graded coursework to address concerns around academic integrity raised in the literature.
6. Prioritize AI tool integration during early debugging and error-detection stages of learning, where the evidence indicates the strongest positive effect, while limiting their role in foundational algorithm-design exercises.

7. Conclusion

This study examined the impact of AI-based tools on programming skill development among 200 university students across three Libyan universities, supported by qualitative insights from 10 programming instructors. The findings indicate that moderate, engaged use of AI tools is associated with meaningful improvements in debugging speed, problem-solving ability, and code efficiency, while excessive reliance may weaken the depth of conceptual understanding that programming education seeks to build. These results underscore the importance of

deliberate pedagogical design when integrating AI into technical education, balancing the genuine productivity benefits of these tools against the risk of skill erosion.

Future research should explore longitudinal effects of AI tool use on long-term retention of programming concepts, expand the sample to include postgraduate students and additional institutions across the region, and examine whether tailored AI usage policies measurably affect learning outcomes over multiple academic semesters.

8. Limitations

This study has several limitations. The sample was drawn using convenience sampling from three Libyan universities, which may limit the generalizability of findings to other regional or institutional contexts. Self-reported survey data are subject to recall and social-desirability bias, particularly regarding perceived skill improvement. Additionally, the cross-sectional design captures a single point in time and cannot establish long-term causal effects of AI tool use on programming proficiency. Future studies employing randomized or longitudinal designs would help address these constraints.

References

[1] A. Smith, R. Patel, and J. Nguyen, “Effects of AI-driven code assistants on student performance in introductory programming courses,” J. Educ. Comput. Res., vol. 38, no. 4, pp. 512–530, 2021.

[2] K. Johnson and S. Lee, “AI-enhanced learning platforms and their impact on student engagement and retention,” IEEE Trans. Learn. Technol., vol. 13, no. 2, pp. 145–157, 2020.

[3] M. Carter and L. Wong, “Intelligent tutoring systems in STEM education: A systematic review,” Comput. Educ., vol. 156, p. 103950, 2020.

[4] P. Anderson, “Personalized feedback mechanisms in adaptive learning environments,” Educ. Technol. Res. Dev., vol. 69, no. 3, pp. 887–905, 2021.

[5] S. Vaithilingam, T. Zhang, and W. Yew, “Expectation vs. experience: Evaluating the usability of code generation tools powered by large language models,” in Proc. CHI Conf. Hum. Factors Comput. Syst. Ext. Abstracts, 2022, pp. 1–12.

[6] J. Denny, S. Prather, and B. Becker, “Conversational AI as a programming learning aid: Effects on conceptual understanding,” in Proc. ACM Conf. Innov. Technol. Comput. Sci. Educ., 2023, pp. 234–240.

[7] D. Miller and Y. Zhou, “The double-edged sword of AI assistance: Risks to critical thinking in technical education,” J. Comput. Assist. Learn., vol. 35, no. 6, pp. 789–802, 2019.

[8] R. Kumar and A. Fitzgerald, “Academic integrity in the age of AI code generation,” ACM Trans. Comput. Educ., vol. 22, no. 3, pp. 1–18, 2022.

[9] H. Park and E. Ahn, “Cognitive load implications of AI-assisted learning in novice programmers,” Comput. Hum. Behav., vol. 128, p. 107098, 2022.

[10] F. Al-Hassan and N. Ibrahim, “Adoption of AI-based educational technologies in Arab higher education institutions,” Int. J. Emerg. Technol. Learn., vol. 17, no. 8, pp. 45–60, 2022.

[11] M. Bouzid and A. Trabelsi, “AI integration in North African engineering curricula: Opportunities and barriers,” in Proc. Int. Conf. Eng. Educ., 2023, pp. 98–106.

[12] T. Becker, “Measuring debugging efficiency in AI-supported integrated development environments,” J. Syst. Softw., vol. 198, p. 111598, 2023.

[13] Alnnale, T. (2026). Predictive Governance in Digital Enterprises: An LSTM-Enhanced Deep Learning Framework for Economic Optimization of IT Incident Management Using Enriched Process Logs. *Al-Farooq Journal of Sciences*, *2*(3), 86-113.

[14] Hadaga, H. M. S. (2026). Exploring Libyan EFL Teachers' Views towards Using Artificial Intelligence Tools in Teaching English Language Translation: A Study at the Department of Translation at the Faculty of Languages within the University of Benghazi. *Al-Farooq Journal of Sciences*, *2*(3), 351-360.