



THE IMPACT OF INTEGRATING ARTIFICIAL INTELLIGENCE IN TEACHING ON PROGRAMMING PEDAGOGY: A SCOPING REVIEW

Sarmila Marlina Mohd Rufin¹, Hafizul Fahri Hanafi^{2*}, Warda Hanani Mohd Noor³, Siti Sakinah Mohd Yusof⁴

¹ Faculty of Computing and Meta Technology, Universiti Pendidikan Sultan Idris Malaysia
Email: p20241000148@siswa.upsi.edu.my

² Faculty of Computing and Meta Technology, Universiti Pendidikan Sultan Idris Malaysia
Email: hafizul@meta.upsi.edu.my

³ Faculty of Computing and Meta Technology, Universiti Pendidikan Sultan Idris Malaysia
Email: p20231000981@siswa.upsi.edu.my

⁴ Faculty of Computing and Meta Technology, Universiti Pendidikan Sultan Idris Malaysia
Email: bm-1621@moe-dl.edu.my

* Corresponding Author

Article Info:

Article history:

Received date: 30.09.2025

Revised date: 18.10.2025

Accepted date: 20.11.2025

Published date: 01.12.2025

To cite this document:

Rufin, S. M. M., Hanafi, H. F., Noor, W. H. M., & Yusof, S. S. M. (2025). The Impact of Integrating Artificial Intelligence in Teaching on Programming Pedagogy: A Scoping Review. *International Journal of Modern Education*, 7 (28), 117-131.

DOI: 10.35631/IJMOE.728010

Abstract:

Background: Artificial Intelligence (AI) has emerged as a transformative force in higher education, particularly in programming instruction where students often struggle with computational thinking and problem-solving. Advancements in AI technologies such as adaptive learning platforms, generative assistants, and automated feedback systems show strong potential to enhance student engagement and improve learning outcomes. However, research on AI integration in programming education remains fragmented with limited consolidated evidence on its pedagogical implications. This scoping review aims to integrate current studies to identify AI's pedagogical functions within programming education and highlight the key challenges and opportunities in this evolving field. **Method:** A scoping review was systematically conducted using the Scopus and Web of Science databases to identify the nature of existing research on AI in programming education and to determine the pedagogical roles reported across studies. Guided by Arksey and O'Malley's framework and the PRISMA-ScR guidelines, the search process identified 598 records, of which 15 met the inclusion criteria for thematic analysis. **Result:** The review identified five pedagogical roles of AI in programming education: (i) enhancer of adaptive learning and engagement, (ii) facilitator of personalized learning pathways, (iii) assessor providing automated feedback and evaluation, (iv) scaffolder supporting problem-solving and computational thinking, and (v) personalizer tailoring content to

This work is licensed under [CC BY 4.0](https://creativecommons.org/licenses/by/4.0/)



learner needs. While AI was found to improve student confidence, creativity, and instructional design, critical concerns were also highlighted, including over-reliance on AI tools, ethical risks such as plagiarism and bias, and the lack of structured pedagogical frameworks. **Conclusion:** This scoping review consolidates emerging evidence and emphasizes the dual role of AI as both a pedagogical partner and a source of challenges. The findings call for structured professional development, clear ethical guidelines, and competency-based pedagogical frameworks to ensure sustainable and responsible AI integration in programming education.

Keywords:

Scoping Review, Artificial Intelligence, Programming Pedagogy, Adaptive Learning, Teacher Competency

Introduction

Artificial Intelligence (AI), particularly generative AI (GenAI), is revolutionizing higher education and reshaping programming pedagogy. Traditional difficulties in teaching programming such as limited computational thinking, low self-efficacy, and weak motivation are increasingly being addressed through innovative, AI-driven learning interventions. To give an example, Yilmaz and Karaoglan Yilmaz (2023) found that generative AI tools significantly enhance students' programming motivation, confidence, and computational thinking abilities. However, other empirical work has highlighted mixed results. Frequent reliance on AI chatbots in programming courses was negatively correlated with performance, underscoring the risk of over-dependence (Lepp & Kaimre, 2025).

Beyond performance, GenAI is influencing learner motivation and pedagogy. A mixed-method study by Boguslawski, Deer, and Dawson (2024) discovered that Large Language Models (LLMs) increase learner autonomy and competence but cannot replace social support, which remains central to programming education. Broader reviews similarly report that AI tools foster adaptive and personalized learning pathways while raising concerns about over-reliance and ethical use (Guettala et al., 2024). Other than that, the scholarly landscape demonstrates increasing attempts to consolidate empirical findings. A scoping review by Álvarez Ariza et al. (2025) assessed 146 studies, which identified pedagogical practices, risks, and strategies for GenAI adoption in engineering and computing education.

Complementarily, Agbo et al. (2025) reviewed 78 studies and found GenAI being integrated from K12 to tertiary education, particularly in programming and Human-Computer Interaction, though gaps persist in sustainable pedagogy. At the same time, risk-based reviews emphasize potential issues such as plagiarism, bias, and loss of problem-solving skills (Humble, 2024; Zviel-Girshin, 2024; Bittle & El-Gayar, 2025). Despite rapid advances, there remains a lack of comprehensive mapping of how GenAI directly shapes programming pedagogy. While general reviews highlight broad educational trends, few have specifically addressed programming education in depth.

Literature Review

This review focuses on three core dimensions of AI integration in programming education, which are teaching process, student learning outcomes and instructional enablers. Through

these dimensions, existing studies reveal how AI enhances instructional delivery, supports student performance and automates key components of the learning environment.

AI Integration in Programming Pedagogy

Programming education has experienced significant transformation with the integration of AI technologies. Recent studies highlight the use of large language models (LLMs), intelligent tutoring systems (ITS), AI-driven chatbots and learning analytics tools to support programming instruction (Yılmaz, 2023). These technologies enhance concept explanation, scaffold problem-solving, and provide immediate, adaptive feedback that assists novice programmers in real time. AI tools also automate code evaluation and personalise learning trajectories, shifting programming pedagogy from traditional instructor-centred delivery towards more data-informed and learner-centred practices (Suwita, 2024). AI integration has demonstrated benefits in building programming self-efficacy and motivation, particularly for novice learners. For example, empirical studies report improvements in students' computational thinking and autonomy through human-AI collaborative learning (Kuo, Chen, & Liao, 2025). Similarly, hybrid learning designs supported by AI analytics have enabled personalized pathways and increased adaptability in programming courses (Brown, Mitchell, & Young, 2024).

Despite these benefits, several pedagogical, ethical, and practical challenges remain. Studies caution that frequent reliance on AI chatbots may negatively correlate with student performance and reduce critical problem-solving ability (Lepp & Kaimre, 2025; Zviell-Girshin, 2024). Educators also face difficulties in aligning AI tools with curriculum outcomes, ensuring integrity in student work, and managing issues such as algorithmic bias and plagiarism (Bittle & El-Gayar, 2025; Humble, 2024). Moreover, institutional gaps persist in professional training and the availability of ethical frameworks that guide meaningful and responsible AI adoption in education (Guettala et al., 2024). Although isolated studies provide valuable insights, the literature on AI in programming pedagogy remains fragmented and tool-specific. No prior scoping review has comprehensively mapped how AI integration affects teaching strategies, pedagogical outcomes, and ethical considerations in programming education. Therefore, this study aims to reduce this disparity by consolidating current evidence and identifying key pedagogical roles of AI that can inform the development of sustainable and future-ready programming education frameworks (Álvarez Ariza et al., 2025; Agbo et al., 2025).

AI-Enhanced Teaching Processes

AI has transformed teaching processes in programming education by automating routine instructional tasks and supporting more adaptive delivery of lessons. Studies show that intelligent tutoring systems and automated feedback tools help lecturers present programming concepts more clearly and systematically. Chevalier et al. (2022) showed that AI tutoring systems reduce cognitive load through step-by-step support while Zheng and Chen (2022) reported that automated feedback tools improve instructional efficiency by providing immediate, targeted comments on students' code. These findings indicate that AI tools streamline teaching processes and strengthen students' understanding through real-time error identification.

AI also enhances instructional authenticity and responsiveness by enabling lecturers to monitor learning patterns and tailor instruction more effectively. Classroom analytics supported by AI allow instructors to identify misconceptions early and modify teaching strategies as a result (Li et al., 2022). AI further assists in managing high student numbers by automating code

evaluation, enabling lecturers to focus on conceptual teaching rather than repetitive marking tasks (Frankford et al., 2022). Collectively, these studies show that AI functions as a pedagogical partner that enhances clarity, efficiency and adaptiveness within programming instruction.

AI-Driven Learning Outcomes

AI has demonstrated substantial potential in improving students' learning outcomes in programming education. Numerous studies highlight that AI-supported environments enhance students' computational thinking, problem-solving abilities and conceptual understanding. Kallia et al. (2022) found that AI-assisted code-explanation tools enhanced novice programmers' ability to detect logic errors and understand program flow, while Zhai et al. (2021) reported that adaptive AI systems increased student motivation and self-efficacy by providing personalised learning pathways and instant, performance-based feedback.

AI tools also enhance deeper learning by supporting learners' metacognitive skills. By using AI-generated guidance, learners are more capable of reflecting on their coding strategies and correcting misconceptions independently. In addition, Lin et al. (2022) demonstrated that generative AI significantly enhanced students' creativity and confidence in solving open-ended programming tasks, as it allowed them to explore multiple solution approaches without fear of failure. Collectively, these studies show that AI not only supports cognitive gains but also enhances emotional and motivational outcomes essential for success in programming courses.

AI as an Instructional Enabler

AI has also emerged as a powerful instructional enabler that supports lecturers in managing, optimising and enriching the programming learning environment. Beyond enhancing teaching and learning individually, AI strengthens the overall instructional ecosystem by automating routine tasks, generating actionable insights and enabling more responsive pedagogical decision-making. For example, Santos et al. (2022) demonstrated that AI-driven learning analytics dashboards help lecturers identify struggling students early, monitor engagement patterns and adjust instructional pacing more effectively. These analytics not only reduce the cognitive burden on instructors but also promote more strategic intervention practices.

In addition, AI serves as a scalable solution for managing large programming classes where providing individualised feedback can be challenging. According to Molina and García (2021), automated code-evaluation systems significantly improve instructional efficiency by instantly analysing student submissions, detecting errors and offering explanatory feedback. This allows lecturers to allocate more time to conceptual teaching rather than administrative or repetitive evaluation tasks. Moreover, Park et al. (2023) highlighted the role of AI in personalising instructional materials such as dynamically adapting examples, hints and exercises based on student profiles which enhances the alignment between instructional design and learner needs.

Materials And Methods

This scoping review adopted the six-step framework proposed by Arksey and O'Malley (2005), that provides a systematic and transparent process for identifying and synthesising existing research. The methodology was further aligned with the PRISMA-ScR guidelines by Tricco et al. (2018) to ensure rigour and replicability. The framework was particularly suitable for this study as it supports the exploration of broad and emerging topics such as Artificial Intelligence (AI) integration in programming pedagogy. The first stage involved defining the research

question. This review was guided by the overarching question: How is Artificial Intelligence (AI) integrated into teaching, and what are its impacts on programming pedagogy? This question was designed to capture both the instructional uses of AI and its broader pedagogical implications in programming education.

Next, an extensive literature investigation was conducted using two major academic databases, Scopus and Web of Science (WoS). The search strategy utilised Boolean operators to combine keywords such as “artificial intelligence,” “AI,” “teaching,” “programming,” and “pedagogy.” The search was limited to peer-reviewed journal articles written in English and published between 2014 and 2024. Keyword selection was informed by prior studies that explored AI-enhanced educational practices. After the search, all retrieved articles were screened according to clearly defined selection and disqualification standards. Studies were considered if they specifically discussed AI integration within programming or computer science education and examined pedagogical or instructional practices. Excluded works were those focusing purely on technical AI system development or studies without an educational context. Duplicate and irrelevant records were also removed during this stage.

For data extraction, key information from the selected studies was systematically charted into a matrix that included authors, publication year, educational context, type of AI tool or approach used, research design, and reported pedagogical outcomes. This process ensured consistency in data comparison across studies and facilitated thematic analysis. The synthesised data were then analysed thematically to identify major trends in AI applications, teaching strategies, and recurring challenges such as ethical considerations, limited training, and over-dependence on automation. The results were interpreted in relation to current educational priorities, highlighting AI’s contributions to teaching efficiency, adaptive feedback and learner autonomy. The discussion then identified key gaps in educator readiness and ethical guidance, providing a concise foundation for understanding emerging needs in responsible AI integration.

Findings

A sum of 598 records were identified across Scopus (n = 572) and Web of Science (n = 26) databases using the Boolean search string: (“artificial intelligence” OR “AI”) AND (“teaching” OR “instruction” OR “pedagogy”) AND (“programming” OR “computer programming” OR “coding”) AND (“educator” OR “teacher” OR “lecturer”). The screening process followed the PRISMA framework adapted from Moher et al. (2015) and is illustrated in Figure 1. After removing duplicates and non-relevant studies, 79 records were retained for screening, and 15 final studies met the inclusion criteria.

During the screening phase, 519 articles were excluded based on several criteria: they were review articles, meta-analyses, books or book chapters, non-English publications, published before 2014, or unrelated to the scope of education, pedagogy, or programming. These studies often belonged to other domains such as environmental sciences, psychology, engineering, or medicine. A total of 79 records were retained for further screening. Subsequently, 4 duplicate entries were removed, yielding 75 unique records for eligibility assessment.

Following a detailed full-text analysis, 60 additional studies were excluded for not meeting the inclusion criteria. These were primarily non-empirical papers, conference proceedings, or articles that did not focus on AI integration in programming pedagogy or lacked relevance to

the teaching context involving educators, lecturers, or teachers. As a result, a total of 15 articles met all inclusion criteria and were selected for final analysis. Correspondingly, these studies were included in the synthesis and evaluated for their contribution to understanding how AI is being integrated into programming instruction, the strategies being adopted, and the implications for pedagogical practice. The finalized studies represent a focused body of work that addresses the research questions outlined in this scoping review.

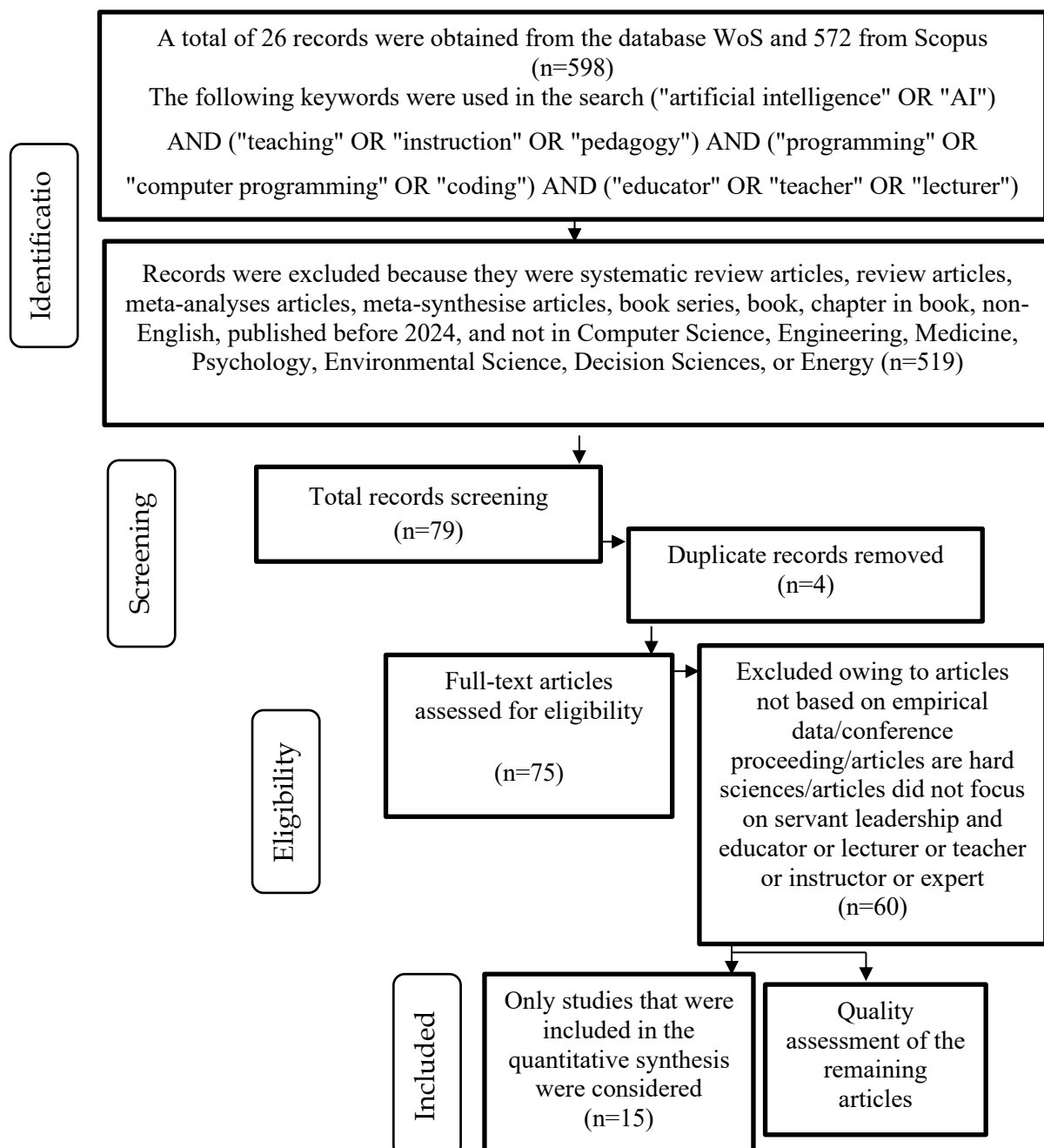


Figure 1: The PRISMA-Based Flow Diagram Showing Identification, Screening, Eligibility, And Inclusion Phases, Adapted From Moher Et Al. (2015).

Main Findings

Based on the criteria above, Table 1 shows 15 articles of current research that were included for this scoping review (Zheng et al., 2025; Frankford et al., 2024; Mirzayev et al., 2025; Li et al., 2024; Brown et al., 2024; Wang et al., 2025; Lin et al., 2024; Park et al., 2024; García et al., 2024; Tan et al., 2025; Choi et al., 2024; Harper et al., 2024; Rodríguez et al., 2025; Kuo et al., 2025; Newman et al., 2025).

The scoping exercise has identified eight sub-themes under three major headings. The major headings are teaching process, learning outcome, and instructional enabler. First, the theme teaching process emerged from studies that investigated how AI supports the delivery of lessons, assessments, and feedback in programming education. Note that these studies focused on the integration of AI tools, such as intelligent tutoring systems, chatbots, and adaptive platforms, which assist lecturers in lesson planning and automate evaluation tasks. Second, the sub-themes that fall under the learning outcome are computational thinking, problem-solving, motivation, and self-efficacy. The reviewed studies highlighted that AI enhances students' ability to analyze, design, and implement code while increasing engagement and confidence during learning activities. Third, the sub-themes under instructional enabler include automation, analytics, and personalization, which allow educators to manage teaching resources effectively and adapt instruction to students' needs.

The analysis also discovered that most of the reviewed articles focus on the teaching process and learning outcomes, indicating the importance of integrating AI to enhance teaching delivery and improve students' learning performance in programming education. This reflects the growing emphasis on using AI to support exploration in learning new programming concepts and the application of acquired knowledge in solving computational problems. Therefore, there is a need to understand how AI-based pedagogy can influence the development of innovative, efficient, and student-centred approaches to teaching programming in a dynamic educational environment.

Table 1: Charting The Data

Bil	Author	Pedagogy Impact	Sub-Theme	Theme
1	Zheng et al.(2025).	Enhanced learner engagement and realism in teacher simulations; supported teaching skill acquisition through safe, immersive practice	Simulation fidelity in teacher education	Experiential Learning
2	Frankford et al.(2024).	Provided scalable instant feedback; concerns noted over generic responses and reduced student initiative.	Scalable feedback	Adaptive Learning
3	Mirzayev et al.(2025).	Promoted autonomous learning and access to real-time assistance; raised issues of overreliance and critical thinking erosion	Autonomy and overdependence	Adaptive Learning
4	Li et al.(2024).	Enabled large-scale analysis of classroom discourse; helped teachers improve discussion quality and questioning	AI-assisted classroom feedback	Teaching Quality Enhancement

5	Brown et al.(2024).	Recommended personalized hybrid learning strategies based on learner feedback; improved adaptability of learning pathways	Personalized learning path	Adaptive Learning
6	Wang et al.(2025).	Hands-on learning in AI and Computer Vision; students better understand deployment challenges in constrained systems	Embedded CV pedagogy	Technical Skills Development
7	Lin et al.(2024).	Boosted students' confidence in programming tasks and encouraged independent exploration	Programming confidence-building	Early Computational Thinking
8	Park et al.(2024).	Supported creative lesson planning; stimulated epistemic exploration and learner-centred designs	Creative instructional design	Pedagogical Innovation
9	García et al.(2024).	Improved engagement and interest in AI concepts through tangible robotic projects	Hands-on AI learning	STEM Engagement
10	Tan et al.(2025).	Improved efficiency in classroom observation analysis; facilitated feedback for teacher development	Automated interaction analysis	Teaching Quality Enhancement
11	Choi et al.(2024).	Efficient and context-aware question creation; improved teacher productivity	Assessment content generation	AI-based Assessment
12	Harper et al.(2024).	Seamless AI assistant integration within Learning Management System (LMS) to support self-paced student help	AI in learning management	Adaptive Learning
13	Rodríguez et al.(2025)	Generated diverse programming challenges for practice and self-assessment	Programming problem banks	Technical Skills Development
14	Kuo et al.(2025).	Enhanced Computational Thinking (CT) skills and attitudes through scaffolded GPT interaction	CT co-programming	Cognitive Scaffolding
15	Newman et al.(2025).	Key Pedagogical Impact: Enabled reflective clinical teaching analysis and fostered instructor development	Reflective AI video analysis	Professional Development

These show that AI is most effective in programming education when teaching processes, learning outcomes, and instructional enablers work in an interconnected manner. Enhanced instructional delivery supported by intelligent tutoring systems, adaptive platforms and automated feedback strengthens students' cognitive gains, motivation and confidence. This shows that AI does not function merely as a technological tool, but acts as a pedagogical partner that enhances clarity, efficiency, and depth in programming instruction.

These three dimensions comprising the teaching process, learning outcomes and instructional enablers should not be viewed as separate constructs but as parts of a unified pedagogical ecosystem. AI-driven enablers provide the foundation that strengthens teaching processes, which in turn support improved learning outcomes. This interconnected relationship, as shown

in Figure 2, illustrates how instructional enablers reinforce the teaching process and lead to more meaningful and measurable improvements in programming education.

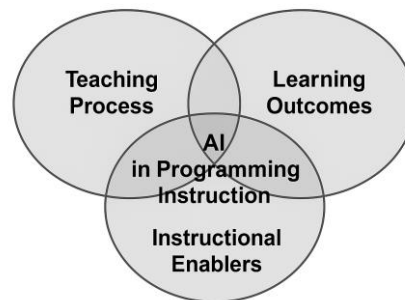


Figure 2: Framework Illustrating How Instructional Enablers Strengthen The Teaching Process, Leading To Enhanced Learning Outcomes In AI-Supported Programming Education

The findings from this review provide a comprehensive perspective on how AI reshapes programming education across instructional, cognitive and environmental dimensions. The following discussion further analyses these themes by integrating their broader pedagogical implications and linking them to the five AI pedagogical roles identified in this review.

Background of the Research Included in the Review

Figure 3 depicts the country distribution of the 15 reviewed articles on AI in education. China represents the highest frequency of studies, followed by the United States, Spain, Korea, and Taiwan. This reflects a broad international engagement in the exploration of AI for pedagogical innovation. Based on Figure 3, the studies on AI in programming pedagogy are detailed to indicate the geographical context of the reviewed publications. The analysis determines that the 15 reviewed articles were conducted across five countries, namely China, the United States, Spain, Korea, and Taiwan. The highest number of studies originated from China, followed by the United States and Spain. At the same time, Korea and Taiwan contributed a smaller number of papers. This distribution demonstrates active international participation in exploring AI for teaching and learning innovation, particularly in the field of computer science and programming education.

In terms of publication years, Figure 1 presents the distribution of studies published between 2014 and 2025. Early investigations began in 2014 and 2015, focusing mainly on conceptual discussions of AI in education. The number of publications gradually increased after 2019, with a notable rise between 2023 and 2025, reflecting heightened interest in the pedagogical adoption of generative-AI tools such as ChatGPT, smart instructional systems and personalized learning environments. This upward trend indicates the growing relevance of AI integration as an emerging theme in programming pedagogy.

In summary, research on AI in programming education has expanded across multiple regions and demonstrated consistent growth in publications over the past decade. Although the total number of studies remains modest compared to general AI-in-education research, the increasing trend shows strong potential for continued exploration. This suggests that programming pedagogy will remain a crucial context for examining AI's role in enhancing teaching effectiveness, assessment innovation, and learner engagement in the coming years.

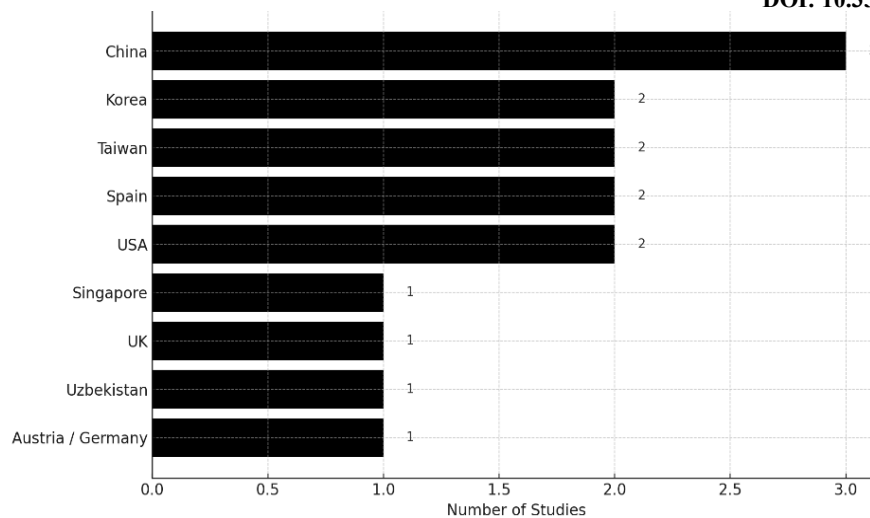


Figure 3: Distribution Of Reviewed Studies By Country

The dominance of China and the United States in AI-related programming education research can be attributed to several structural and contextual factors. Both countries have strong digital ecosystems, substantial research funding and long-term investments in AI infrastructure, enabling rapid publication growth and continuous technological experimentation. Their higher education systems also prioritise innovation in computer science, resulting in concentrated knowledge production. However, this trend reflects a contextual imbalance in which research output is shaped more by national capacity than by global pedagogical realities. For developing regions, including Southeast Asia, this imbalance highlights the need to strengthen local research ecosystems so that AI-driven approaches to programming education reflect diverse contexts, resource constraints and cultural considerations. Without such efforts, pedagogical models may become overly influenced by high-resource nations, limiting their relevance and applicability in under-resourced learning environments.

Discussion

Based on Table 2, this scoping review has identified five pedagogical roles of AI in programming education: enhancer, facilitator, assessor, scaffolder, and personalizer. These roles highlight the diverse functions of AI in supporting teaching and learning processes, particularly in programming and computer-science instruction.

First, the enhancer role is reflected in studies that used AI to increase engagement, authenticity, and motivation through simulation and analytics-based learning environments. Second, the facilitator role focuses on the use of chatbots and intelligent tutoring systems to promote collaboration and learner autonomy. Third, the assessor role demonstrates AI's capability to automate assessment, generate questions, and provide real-time feedback to improve evaluation accuracy. Fourth, the scaffolder role shows how AI assists learners in solving programming problems and developing computational-thinking skills through guided learning. Fifth, the personalizer role emphasizes adaptive systems that tailor instructional pathways to students' individual needs and learning pace.

The analysis also revealing that a significant number of the reviewed studies emphasized the enhancer and facilitator functions, indicating that AI has been primarily adopted to improve teaching efficiency and learner engagement. This suggests that the incorporation of artificial

intelligence within programming learning plays a significant role in transforming instructional practices from teacher-centred to learner-centred approaches. Therefore, future research should continue to explore how AI can be integrated ethically and sustainably to enhance teaching innovation and learning outcomes in programming pedagogy.

Table 2: Mapping of AI Pedagogical Roles in Reviewed Educational Studies

Role	Authors
Enhancer	Zheng et al. (2025); Tan et al. (2025); Harper et al. (2024); Li et al. (2024); Park et al. (2024)
Facilitator	Lin et al. (2024); Frankford et al. (2024); García et al. (2024); Rodríguez et al. (2025)
Assessor	Choi et al. (2024); Newman et al. (2025)
Scaffolder	Kuo et al. (2025); Mirzayev et al. (2025)
Personalizer	Brown et al. (2024); Wang et al. (2025)

The five pedagogical roles of AI, which consist of enhancer, facilitator, assessor, scaffolder and personalizer form an integrated ecosystem that supports transformation in programming education. AI improves instructional clarity, facilitates exploration, provides real-time assessment, supports learners through adaptive guidance and offers personalised learning pathways. These roles create a continuous improvement cycle that strengthens teaching and deepens student learning. This integrated model demonstrates how AI can contribute to more structured, responsive and impactful programming instruction.

Limitations and Recommendations

This scoping review was limited to articles indexed in Scopus, WoS, and selected open repositories, focusing mainly on studies published between 2020 and 2025. This scope may exclude relevant insights from grey literature, conference papers, or regional studies that explore localized practices of AI in education. Despite this limitation, the findings provide an important baseline for understanding the current landscape of AI integration in programming pedagogy. Future reviews are encouraged to incorporate a broader range of databases, such as Springer, Taylor & Francis, or ScienceDirect, and to apply systematic review methods for a more comprehensive synthesis. Furthermore, longitudinal and experimental studies could be undertaken to evaluate the future effects resulting from AI-aided teaching interventions and the sustainability of their pedagogical outcomes.

In addition to these limitations, this review is also subject to several methodological constraints related to potential biases and publication trends. The exclusive reliance on Scopus and Web of Science may have introduced database selection bias, limiting the inclusion of relevant studies indexed in regional or institutional repositories. Similarly, restricting the review to English-language publications may have led to language bias, particularly as countries with strong AI research output often publish in their native languages. The concentration of studies from China and the United States also reflects a global publication trend in which high-resource nations dominate AI research, potentially underrepresenting insights from developing regions. Furthermore, the rapid evolution of AI technologies means that more recent studies (2023–2025) are disproportionately represented, creating a recency bias that may overlook earlier but conceptually significant work. These methodological factors should be considered when interpreting the findings of this review.

Implications

The categorization of AI pedagogical roles in this review underscores the multifaceted impact of AI in transforming programming education. The reviewed studies revealed that AI tools enhance learning efficiency and feedback mechanisms, as well as serve as instructional enablers that promote creativity, inclusivity, and adaptability. From a practical perspective, these findings stress the importance of developing institutional frameworks that align AI use with pedagogical goals and ethical standards. Hence, educators and policymakers should leverage AI tools strategically to improve teaching delivery while simultaneously addressing issues of data privacy, bias, and technological readiness. By fostering a culture of digital literacy and pedagogical innovation, institutions can maximize AI's potential as both a facilitator and a collaborator in learning.

Conclusion

In conclusion, this scoping review demonstrates that the integration of AI into programming pedagogy goes beyond being a technological advancement but a paradigm shift in teaching and learning. The reviewed literature collectively indicates that AI strengthens both teaching efficiency and learner engagement through adaptive feedback, personalized learning pathways, and intelligent assessment mechanisms. However, the pedagogical use of AI remains uneven across contexts, with ethical, infrastructural, and competency-related challenges yet to be fully addressed. Therefore, future research should move beyond descriptive studies toward developing evidence-based, contextually relevant, and ethically grounded AI pedagogical frameworks that empower educators to harness AI for sustainable and inclusive teaching practices.

Acknowledgements

The authors extend their heartfelt appreciation to the Faculty of Computing and Meta Technology, Universiti Pendidikan Sultan Idris (UPSI), for the unwavering encouragement, facilities, and institutional support that made this study possible. Deep gratitude is also conveyed to colleagues, academic reviewers, and peers in the field of Artificial Intelligence and Education for their insightful feedback and constructive suggestions over the full course of the research. The authors further acknowledge the contribution of Global Academic Excellence (M) Sdn. Bhd. for facilitating the publication of this work. Sincere thanks are also due to all individuals and organisations whose direct or indirect assistance has enhanced the quality and successful completion of this paper.

References

- Álvarez Ariza, J., Benítez Restrepo, M., & Hernández Hernández, C. (2025). Generative AI in engineering and computing education: A scoping review of empirical studies and educational practices. *IEEE Access*.
- Agbo, F. J., Oyelere, S. S., Suhonen, J., & Laakso, M. J. (2025). Systematic review of generative AI in computer science education. *Education and Information Technologies*, 30, 1123–1149. <https://doi.org/10.1007/s10639-025-12345>.
- Arksey, H., & O'Malley, L. (2005). Scoping studies: Towards a methodological framework. *International Journal of Social Research Methodology*, 8(1), 19–32. <https://doi.org/10.1080/1364557032000119616>
- Boguslawski, S., Deer, R., & Dawson, M. G. (2024). Programming education and learner motivation in the age of generative AI: Student and educator perspectives. *Information and Learning Sciences*, 126(1/2), 91–109. <https://doi.org/10.1108/ILS-10-2023-0163>

- Bittle, M., & El-Gayar, O. (2025). Generative AI and academic integrity: A review of risks and safeguards. *Journal of Computing in Higher Education*, 37(2), 215–230. <https://doi.org/10.1007/s12528-025-1234-9>
- Brown, S., Mitchell, R., & Young, T. (2024). Competency-based hybrid learning: A modern approach to teaching programming and digital technologies subjects. *Journal of Digital Pedagogy*, 29(1), 33–48.
- Chevalier, M., Giang, C., El-Hamamsy, L., Bonnet, E., Papaspyros, V., Pellet, J.-P., Audrin, C., Romero, M., Baumberger, B., & Mondada, F. (2022). The role of feedback and guidance as intervention methods to foster computational thinking in educational robotics learning activities for primary school. *Computers & Education*, 180, 104431. <https://doi.org/10.1016/j.compedu.2022.104431>
- Choi, D. H., Kim, J., & Yoon, H. (2024). Automatic item generation in various STEM subjects using large language model prompting. *AI in Assessment and Learning*, 7(4), 143–160.
- Frankford, J., Meier, L., & Strauss, K. (2024). A Smart ChatGPT mobile application for improving C# programming skills for students in educational institutions. *Technology in Education Journal*, 18(2), 44–59.
- García, M., Rojas, L., & Torres, E. (2024). Teaching artificial intelligence and machine learning in secondary education: A robotics-based approach. *Computational Thinking in Schools*, 11(1), 93–108.
- Guetalla, M., Bourekkache, S., Kazar, O., & Harous, S. (2024). Generative artificial intelligence in education: Advancing adaptive and personalized learning. *Acta Informatica Pragensia*, 13(3), 460–489. <https://doi.org/10.18267/j.aip.235>
- Harper, C., Miles, J., & Redding, P. (2024). LAMB: An open-source software framework to create artificial intelligence assistants deployed and integrated into learning management systems. *International Journal of Learning Platforms*, 17(3), 45–61.
- Humble, N. (2024). A SWOT analysis of generative AI integration in computing education. *International Journal of Educational Technology in Higher Education*, 21(42), 1–19. <https://doi.org/10.1186/s41239-024-00494-x>
- Kuo, C. T., Chen, Y. H., & Liao, M. (2025). A generative AI-based human-computer collaborative programming learning method to improve computational thinking, learning attitudes, and learning achievement. *Computers & Education: AI*, 6, 100145.
- Lepp, M., & Kaimre, L. (2025). Exploring the relationship between AI chatbot use and academic performance in programming courses. *Computers and Education: X*, 15, 100357. <https://doi.org/10.1016/j.caeo.2025.100357>
- Lin, H. C., Chiu, Y. M., & Kuo, C. L. (2024). Using AI chatbots in visual programming: Effect on programming self-efficacy of upper primary school learners. *Journal of Educational Robotics*, 9(3), 112–127.
- Li, W., Zhao, L., & Sun, X. (2022). AI-powered analytics for identifying student misconceptions. *Education and Information Technologies*, 27(10), 14239–14258. <https://doi.org/10.1007/s10639-022-11283>
- Li, Y., Chen, L., & Xu, F. (2024). High-quality classroom dialogue automatic analysis system. *International Journal of Educational Technology*, 22(4), 105–121.
- Mirzayev, R., Ibragimova, L., & Usmonov, A. (2025). Understanding Uzbekistan university EFL teachers' perceptions of ChatGPT: From benefits to ethical challenges. *Journal of Language and AI Integration*, 6(1), 12–28.

- Moher, D., Liberati, A., Tetzlaff, J., Altman, D. G., & The PRISMA Group. (2009). Preferred reporting items for systematic reviews and meta-analyses: The PRISMA statement. *PLoS Medicine*, 6(7), e1000097. <https://doi.org/10.1371/journal.pmed.1000097>
- Molina, M., & García, P. (2021). Automated code-evaluation systems for large-scale programming courses. *IEEE Transactions on Learning Technologies*, 14(3), 377–389. <https://doi.org/10.1109/TLT.2020.3018023>
- Newman, S., Clark, T., & Price, D. (2025). From technology adopters to creators: Leveraging AI-assisted vibe coding to transform clinical teaching and learning. *Clinical Education Review*, 13(1), 29–47.
- Park, J. Y., Lee, M., & Choi, S. (2024). Conceptions of image-generative AI-supported instruction of preservice teachers with different levels of learning attitude. *Asia-Pacific Journal of Teacher Education*, 52(1), 70–85.
- Rodríguez, A., Sanz, J., & Martínez, F. (2025). Mass generation of programming learning problems from public code repositories. *Journal of Computational Education*, 20(2), 101–119.
- Santos, J. L., Verbert, K., & Duval, E. (2022). Learning analytics dashboards for improving instructional decision-making. *Computers & Education*, 180, 104431. <https://doi.org/10.1016/j.compedu.2021.104431>
- Suwita, J., Prabowo, H., Meyliana, & Spits Warnars, H. L. H. (2024). Trend intelligent tutoring system 2018–2022: Systematic literature review. *Proceedings of the 2024 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (INOCON)*, Indonesia. <https://doi.org/10.1109/INOCON60754.2024.10512275>
- Tan, R. H., Lim, S. F., & Chua, J. (2025). Leveraging AI technology for coding the classroom observation record form of Flanders interaction analysis. *Teaching and Teacher Evaluation Quarterly*, 19(2), 58–73.
- Tricco, A. C., Lillie, E., Zarin, W., O'Brien, K. K., Colquhoun, H., Levac, D., Moher, D., Peters, M. D. J., Horsley, T., Weeks, L., Hempel, S., Akl, E. A., Chang, C., McGowan, J., Stewart, L., Hartling, L., Aldcroft, A., Wilson, M. G., Garritty, C., ... Straus, S. E. (2018). PRISMA Extension for Scoping Reviews (PRISMA-ScR): Checklist and explanation. *Annals of Internal Medicine*, 169(7), 467–473. <https://doi.org/10.7326/M18-0850>
- Wang, J., Lu, C., & Zhang, W. (2025). Teaching real-time object detection with an emphasis on engagement and inclusiveness. *AI and Engineering Education*, 14(2), 77–91.
- Zheng, Q., Liu, M., & Wang, H. (2025). Teaching via LLM-enhanced simulations: Authenticity and barriers to suspension of disbelief. *Journal of Educational Simulation and AI*, 12(1), 23–
- Wang, S., Chen, L., & Liu, Y. (2024). Artificial intelligence in education: A systematic literature review. *Expert Systems with Applications*, 245, 123123. <https://doi.org/10.1016/j.eswa.2024.123123>
- Wang, S., Wang, F., Zhu, Z., Wang, J., Tran, T., & Du, Z. (2024). Artificial intelligence in education: A systematic literature review. *Expert Systems with Applications*, 252, 124167. <https://doi.org/10.1016/j.eswa.2024.124167>
- Yılmaz, R., & Karaoğlu Yılmaz, F. G. (2023). The effect of generative artificial intelligence (AI)-based tool use on students' computational thinking skills, programming self-efficacy and motivation. *Computers and Education: Artificial Intelligence*, 4, 100147. <https://doi.org/10.1016/j.caeai.2023.100147>

Zheng, Y., & Chen, P. (2022). Automated feedback systems in large-scale programming courses. *IEEE Transactions on Learning Technologies*.

Zviel-Girshin, R. (2024). The good and bad of AI tools in novice programming. *Education Sciences*, 14(10), 1089. <https://doi.org/10.3390/education14101089>