Application of Artificial Intelligence in Computer Programming Education

Junhao Su

Zhanjiang University of Science and Technology, Zhanjiang, 524086, China

13824842714@139.com

Abstract. With the rapid development of information technology, the importance of computer programming education is increasingly prominent. This study explores the application of Artificial Intelligence (AI) in computer programming education, analyzing its theoretical foundations, specific practices, challenges, and future directions. The study indicates that AI technology, through tools such as intelligent tutoring systems, automated assessment systems, and virtual laboratories, can provide personalized learning paths, real-time feedback, and safe practice environments, thereby improving students' learning efficiency and teachers' teaching effectiveness. However, the study also identifies challenges in technical implementation, teacher training, student acceptance, and resource costs. Case studies of platforms like the "Xiaoya Platform" at Central China Normal University and "Shuishan Online" at East China Normal University demonstrate the effectiveness and feasibility of AI technology in actual teaching. Future research should focus on interdisciplinary studies, long-term effect evaluations, technological integration and innovation, optimization of personalized learning paths, and resource optimization and allocation.

Keywords: Artificial Intelligence, Computer Programming Education, Technical Practice.

1 Introduction

As information technology rapidly advances, computer programming emerges as an essential skill in modern society, vital for fostering students' logical thinking and problem-solving abilities, and foundational for their career development (Jashari et al., 2023) [1]. However, traditional programming education faces challenges such as unequal resource distribution, repetitive teaching methods, and struggles to meet individual needs, all of which stifle the spread and enhancement of programming skills (Erol et al., 2022) [2]; Morrison et al., 2020 [3]; Boom et al., 2022) [4].

Artificial Intelligence (AI), a pivotal force in the tech revolution, is reshaping educational methodologies by providing tools like intelligent tutoring systems that offer personalized assistance and learning suggestions (Sharma & Harkishan, 2022) [5]. AI's capabilities extend to automating code generation and assessment, significantly boosting efficiency and helping students quickly enhance their programming skills (Duong

J. Yin et al. (eds.), *Proceedings of the 4th International Conference on New Media Development and Modernized Education (NMDME 2024)*, Advances in Intelligent Systems Research 188,
https://doi.org/10.2991/978-94-6463-600-0_49

& Chen, 2024) [6], and tailoring educational content based on analysis of learning data (Su & Zhong, 2022) [7]. Despite its benefits, AI in education faces challenges such as potential quality and security issues in AI-generated code (Gupta et al., 2023) [8], the risk of neglecting fundamental programming knowledge (Dickey et al., 2024) [9], and the limitation on creative thinking due to reliance on established coding practices (Verganti et al., 2020) [10]. This study explores AI's role in programming education, its practical applications, challenges, and future directions.

2 Theoretical Foundations in AI Computer Education

Constructivist Learning Theory posits that learning is an active, constructive process, and AI is revolutionizing this approach by customizing education to the learner's pace and preferences (See Fig.1). In programming education, AI-driven personalized learning paths dynamically adjust content and tasks to suit individual progress and interests, enhancing engagement and skill development. Intelligent tutoring systems provide instant feedback and specialized guidance, recognizing each student's unique error patterns and skill levels. This individualized support enables gradual skill mastery while AI-enhanced virtual labs simulate real-world applications, deeply aligning with constructivist ideals by facilitating the practical application of theoretical knowledge.

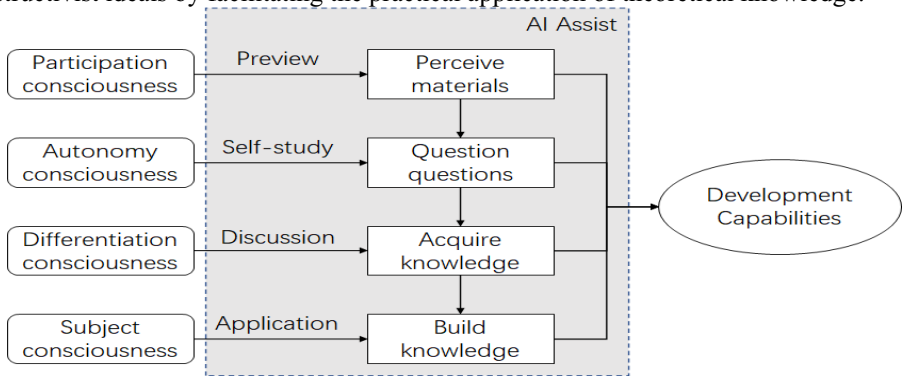

Fig. 1. AI Support for Constructivist Learning Theory

Behaviorist Learning Theory focuses on modifying learning behaviors through external reinforcement, an area where AI greatly contributes by analyzing behavior to customize learning experiences. Intelligent systems adapt problem difficulty and types based on student responses, fostering progress through optimally challenging scenarios. Incorporating gamification and rewards, these systems enhance motivation and engagement. Concurrently, Cognitive Load Theory emphasizes efficient cognitive resource management, with AI playing a crucial role by recommending tailored educational resources and adjusting content to the learner's capacity, thus preventing cognitive overload. These AI applications streamline the learning process, ensuring it is both manageable and tailored to individual cognitive limits and needs.

3 Specific Applications of AI in Programming Education

3.1 Intelligent Tutoring Systems

Intelligent tutoring systems are transforming programming education by providing personalized learning paths and real-time feedback, which enhance learning outcomes significantly. Leveraging AI, these systems analyze student behaviors and data to offer customized recommendations that cater to individual learning needs—unlike traditional methods. For example, beginners might receive guidance on basic concepts and simple exercises, while advanced learners could tackle more complex projects. This targeted approach not only increases learning efficiency but also maintains students' motivation. Additionally, these systems utilize natural language processing and deep learning to offer immediate, precise feedback on coding errors, enhancing skill development. Virtual labs and simulations replicate real-world scenarios, allowing students to actively improve their coding skills. With future integrations of virtual and augmented reality expected, these learning environments promise to become even more immersive, further enhancing practical skills and problem-solving abilities.

3.2 Automated Assessment Systems

Automated assessment systems are revolutionizing programming education by leveraging AI to efficiently evaluate students' programming assignments and exams, thereby improving teaching quality and learning outcomes. These systems reduce teachers' grading workload, especially in large classes, by using predefined standards and algorithms to assess work based on correctness, efficiency, and readability. This automation allows teachers to devote more time to instruction and mentoring. Crucially, the systems provide instant feedback, enabling students to quickly identify and correct errors in their code, which maintains motivation and boosts confidence. Unlike traditional methods that involve delayed feedback, automated systems provide real-time analysis, identifying syntax and logical errors and suggesting improvements. Moreover, these systems offer detailed data analysis, giving teachers insights into students' progress and common challenges, allowing for tailored teaching strategies and supporting educational research. As AI evolves, these systems are expected to gain advanced features like natural language processing and deep learning, enhancing their ability to deliver intelligent and personalized assessments that improve students' programming skills.

3.3 Virtual Laboratories

Virtual laboratories in programming education, powered by AI and virtual reality, simulate real-world environments, providing a safe and flexible space for students to practice programming. These labs prevent typical issues such as system crashes and data loss, allowing students to write, debug, and run code risk-free. They enhance learning by offering real-time monitoring and feedback, which boosts students' skills, confidence, and engagement. Additionally, virtual labs offer unparalleled access flexibility, utilizing cloud computing and remote access technologies to allow students to engage

with programming tools and resources anytime, anywhere. This supports personalized learning experiences that adapt to individual schedules and learning paces, fostering self-directed learning and effective time management. With a wealth of materials and a range of tasks from simple algorithms to complex system development, these labs cater to various skill levels and interests. Looking ahead, the integration of augmented reality (AR) and mixed reality (MR) in these labs promises to create even more immersive and interactive educational experiences.

3.4 Case Studies

Case Selection Criteria. In this section, we analyze two exemplary cases to explore the specific applications of Artificial Intelligence (AI) in computer programming education. The selection of cases was primarily based on the following key criteria: Firstly, educational impact was a core consideration, focusing on platforms that demonstrate significant improvements in teaching quality and learning efficiency through the use of AI, supported by concrete data. Secondly, the cases had to showcase innovation and advancement in AI within programming education, including but not limited to intelligent tutoring systems and automated assessment systems. Additionally, the breadth of implementation and sustainability were also criteria, favoring tools already adopted by multiple educational institutions and capable of ongoing updates, to demonstrate their adaptability and long-term viability. Based on these criteria, this study selected the “Xiaoya Platform” at Central China Normal University and “Shuishan Online” at East China Normal University as the main subjects of research. Since 2019, both platforms have been implemented in several universities across China, reaching over 200,000 student users and achieving considerable educational impact. The platforms enhance the interactivity and personalized learning experiences in programming education by integrating course knowledge graphs, intelligent question-and-answer systems, and personalized recommendation mechanisms. Their successful application not only illustrates the efficacy of AI in educational practice but also provides valuable insights and experiences for further applications of AI in the educational field. Through an in-depth analysis of these cases, we can gain a comprehensive understanding of the actual effects and challenges of AI technology in educational settings, providing a solid foundation for future research and technological development.

Xiaoya Platform. The "Xiaoya Platform" at Central China Normal University is an AI-driven SPOC platform that enhances teaching and learning through integrated course knowledge graphs, intelligent Q&A, and personalized recommendations. It customizes learning paths with knowledge graph technology and machine learning, adjusting content and difficulty based on individual student progress. The platform employs natural lan-guage processing and deep learning for real-time feedback on coding errors, includes a responsive Q&A system to boost motivation, and features virtual laboratories that simulate realistic programming environments for practical skills development. Ad-ditionally, it utilizes comprehensive big data analysis to generate detailed evaluation reports, identifying student weaknesses and optimizing teaching strategies.

Shuishan Online. "Shuishan Online" at East China Normal University is a next-generation AI-powered online learning platform that improves the educational process

from preparation to management. It personalizes learning experiences using knowledge graphs and dynamic content adjustments, and its advanced natural language processing and deep learning capabilities swiftly correct programming errors, enhancing student confidence. The platform's virtual labs provide a secure environment for practicing and refining programming skills anytime, anywhere. Shuishan Online's thorough analysis of learning data produces detailed reports that provide educational insights, helping educators tailor their approaches to effectively meet diverse learner needs.

4 Challenges of AI in Computer Programming Education

4.1 Technical Implementation Difficulties

Applying AI in computer programming education involves multiple challenges, particularly in data handling. AI systems require vast amounts of high-quality data, yet collecting this data is complicated due to privacy issues, the need for diverse data to avoid biases, and the complexities of data preprocessing. Choosing the appropriate AI models is also critical; for example, natural language processing models are ideal for code generation and error detection, whereas machine learning models are more effective for personalizing learning paths. These selections must consider a balance between accuracy, computational demands, and resource usage. Moreover, AI systems need to provide real-time feedback and maintain high processing capabilities to adjust dynamically to students' needs, requiring robust computational support and system stability. Integrating AI into existing educational platforms presents technical challenges in system compatibility and scalability. Additionally, user interface design is crucial for technology adoption, necessitating simplicity and intuitiveness to ensure usability for both students and teachers without adding to their cognitive load. As AI and programming education continue to evolve, regular updates and maintenance of AI systems are essential, demanding a technically proficient team that stays current with technological and educational developments.

4.2 Teacher Training and Adaptation

Integrating AI into computer programming education requires significant updates in teacher training and methodology adaptation. As AI technologies such as intelligent code completion and error detection rapidly evolve, teachers must continually update their knowledge of AI fundamentals to effectively utilize these tools. This necessitates a steep learning curve, particularly for those without a technical background, and requires a shift away from traditional teaching methods towards AI-driven personalized teaching and detailed data analysis. However, effective teacher training encounters challenges like limited resources—time, funding, and specialized materials—and many educational institutions fall short in providing the necessary investment for systematic training tailored to diverse teaching needs. Additionally, teachers' psychological readiness and attitudes towards AI, including concerns about job displacement or reduced teaching quality, significantly influence technology adoption. Overcoming these challenges with regular, personalized training, sufficient resource allocation, psychological

support, and hands-on experiences is crucial for enhancing teacher adaptability and promoting the widespread use of AI in programming education.

4.3 Student Acceptance and Adaptability

The introduction of AI technology in programming education presents various challenges for students, particularly with the complexity of AI-assisted tools that require significant effort to master and continual updates to stay current with advancements. While AI offers personalized learning experiences, student acceptance varies; some prefer traditional methods over AI-driven customization, necessitating strong self-directed learning and time management skills. Psychological and emotional responses also significantly impact student receptivity to AI. Concerns about AI replacing human teachers and changing traditional dynamics can deter students from adopting these technologies, with trust issues regarding AI's reliability further exacerbating resistance and potentially hindering learning progress. Additionally, socioeconomic factors influence engagement with AI tools, as students from less affluent backgrounds may lack access to necessary resources. The level of institutional and community support plays a critical role in enhancing or impeding students' adaptability to AI technology in education.

4.4 Resources and Costs

Implementing AI technology in educational settings requires substantial investments in robust computing infrastructure and stable network environments, posing challenges for resource-limited schools. Essential upgrades include costly hardware like servers and network systems, and ongoing expenses such as licenses, subscriptions, and maintenance for AI-driven tools like programming applications and intelligent teaching systems. These financial demands are compounded by the need for continual software updates, often representing a significant strain, especially when educational funding is scarce. Effective AI integration also necessitates comprehensive teacher training, involving costs for expert instructors, training materials, and organizing sessions, along with potential disruptions to regular teaching schedules. Additionally, maintaining AI systems requires constant technical support and periodic upgrades, possibly involving additional costs for hiring staff or outsourcing. Ensuring data security through advanced software and expert personnel further escalates the costs, highlighting the extensive resource investment needed for a secure and effective implementation of AI technology in education.

5 Conclusion

This study delves into the application of AI in computer programming education, exploring its theoretical basis, practical uses, and prevalent challenges. It demonstrates that AI-driven tools like intelligent tutoring systems, automated assessment systems, and virtual laboratories significantly improve educational outcomes by providing personalized learning experiences, instant feedback, and secure practice environments.

Highlighted case studies, including the “Xiaoya Platform” at Central China Normal University and “Shuishan Online” at East China Normal University, attest to AI's effectiveness in real-world educational settings. Despite its potential, the adoption of AI faces hurdles such as technical complexities, teacher training needs, student resistance, and financial constraints. Addressing these challenges calls for a holistic strategy involving enhanced data protection, teacher training in AI, optimized student experiences, strategic resource allocation, and robust evaluation mechanisms. Future research should focus on interdisciplinary applications, assessing long-term effects, and technological innovations to refine AI integration in education and ensure its sustainable impact.

References

1. Jashari, X., Fetaji, B., & Guetl, C. (2023, August). Assessment of Digital Programing Skills based on the Competencies Model. In 2023 International Conference on Computing, Electronics & Communications Engineering (iCCECE) (pp. 157-162). IEEE.
2. Erol, O., & Çırak, N. S. (2022). The effect of a programming tool scratch on the problem-solving skills of middle school students. Education and Information Technologies, 27(3), 4065-4086.
3. Morrison, C., Villar, N., Thieme, A., Ashktorab, Z., Taysom, E., Salandin, O., ... & Zhang, H. (2020). Torino: A tangible programming language inclusive of children with visual disabilities. Human–Computer Interaction, 35(3), 191-239.
4. Boom, K. D., Bower, M., Siemon, J., & Arguel, A. (2022). Relationships between computational thinking and the quality of computer programs. Education and information technologies, 27(6), 8289-8310.
5. Sharma, P., & Harkishan, M. (2022). Designing an intelligent tutoring system for computer programing in the Pacific. Education and Information Technologies, 27(5), 6197-6209.
6. Duong, H. T., & Chen, H. M. (2024). ProgEdu4Web: An automated assessment tool for motivating the learning of web programming course. Computer Applications in Engineering Education, e22770.
7. Su, J., & Zhong, Y. (2022). Artificial Intelligence (AI) in early childhood education: Curriculum design and future directions. Computers and Education: Artificial Intelligence, 3, 100072.
8. Gupta, M., Akiri, C., Aryal, K., Parker, E., & Praharaj, L. (2023). From chatgpt to threatgpt: Impact of generative ai in cybersecurity and privacy. IEEE Access.
9. Dickey, E., Bejarano, A., & Garg, C. (2024). AI-Lab: A Framework for Introducing Generative Artificial Intelligence Tools in Computer Programming Courses. SN Computer Science, 5(6), 720.
10. Verganti, R., Vendraminelli, L., & Iansiti, M. (2020). Innovation and design in the age of artificial intelligence. Journal of product innovation management, 37(3), 212-227.