International Conference on Social Science and Humanity
Homepage : https://proceeding.antispublisher.or.id/index.php/icossh
Email : admin@antispublisher.com

e-ISSN : 3063-9611
ICOSSH, Vol. 2, No. 1, 2025
Page 786-789
© 2025 ICOSSH

Application of Artificial Intelligence in Digital Information Technologies and Programming Disciplines in Higher Educational Institutions

Khudoyberdiev Abdumalik Dilmurodovich
Samarkand Branch of the ISFT Institute, Uzbekistan

Check for updates OPEN ACCESS
(cc) BY-SA

DOI : https://doi.org/10.61796/icossh.v2i1.226

Sections Info

Article history:
Submitted: January 31, 2025
Final Revised: February 28, 2025
Accepted: March 17, 2025
Published: March 31, 2025

Keywords:
Artificial intelligence
Programming
Coding
Project management
Machine learning
Educational technologies

ABSTRACT

Objective: *The application of artificial intelligence and digital technologies in the educational process is becoming one of the main directions of the modern education system. This document discusses the technologies for using artificial intelligence in teaching programming, their integration into the educational process, as well as tools that increase efficiency for students.* ***Method:*** *First of all, it is emphasized that artificial intelligence platforms for automating code writing, such as GitHub Copilot or ChatGPT, serve to create an easy and fast programming environment for students. The importance of project management tools in developing students' teamwork skills in education is considered. The process of creating machine learning models using AI and applying them to solve practical problems is also covered in detail.* ***Results:*** *It is noted that the role of AI technologies is strong in making the educational process interactive, personalized and innovative.* ***Novelty:*** *The document presents modern resources, platforms, software tools and scientific sources recommended for students. This annotation helps to understand the role of artificial intelligence in teaching digital technologies and programming.*

INTRODUCTION

Artificial intelligence (AI) significantly simplifies the process of learning digital technologies and programming. Below is a complete description of the main areas of teaching programming based on artificial intelligence and the advantages of using them.
1. Automating code writing AI technologies support students in the programming process and simplify code writing. This allows beginners to learn programming faster
Popular tools:
• GitHub Copilot:
Works as an AI assistant for coding.
This tool automatically suggests code by predicting the functions that students want to write.
Helps beginners understand complex algorithms or programs.
• ChatGPT and Codex:
Students get quick answers by asking questions to solve programming problems.
Helps them fix, improve, or explain a specific software project.
Advantages:
• Increases the speed of coding.
• Detects errors in advance and provides advice on how to fix them.
• Helps them learn new programming languages faster.
Practical applications:

• Students will be able to automate the process of developing complex algorithms, such as data structures or software solutions.

• AI, acting as a mentor, will help students independently solve various coding problems.

2. Project management in the process of teaching programming, it is important for students to work on team projects. Artificial intelligence tools help improve project management and efficiency.

AI-powered project management tools:

• Trello and Monday.com:

Used for task planning and distribution.

AI can be used to automatically assign important tasks and monitor their execution.

• Jira and Asana:

Helps students track project progress, manage tasks, and ensure timely completion in teamwork.

• AI-powered code integration:

Automated testing and CI/CD (Continuous Integration/Continuous Deployment) systems through platforms like GitHub help students test their knowledge in practice while managing code repositories.

Advantages:

• Develops students' teamwork skills.

• Makes it easier to plan projects, set deadlines, and evaluate results.

• Teaches students how to work with real-life software projects.

Practical applications:

• Students plan project phases during the development of software applications.

• Practices organizing projects and effectively allocating resources.

3. Teaching Programming with AI

The use of AI technologies in programming lessons teaches students how to create artificial intelligence models and use them to solve real-life problems.

Ways to learn to program with AI technologies:

• Building AI models:

Students learn to build artificial intelligence and machine learning models using libraries such as TensorFlow, PyTorch, or Scikit-learn.

• Data analysis:

Students learn to use the Python or R programming languages to collect, clean, and analyze large amounts of data.

• Natural language processing (NLP):

Using AI to create chatbots or develop automatic translation systems.

• Computer vision:

Creating algorithms for image recognition, object detection, and video stream processing.

• Deep learning:

Learning artificial neural networks (ANN, CNN, RNN) in the programming process.

Advantages:

• Develops important practical skills in the field of artificial intelligence in students.
• Teaches how to apply algorithms to solve real-world problems.
• Prepares students for future technologies and the job market.

Practical application:

• Students can create recommendation systems or predictive models during the learning process.
• Acquires key skills to work on real-world projects in e-commerce, medicine, or technology.

RESEARCH METHOD

This study adopts a qualitative descriptive approach to explore the implementation of artificial intelligence (AI) in programming education. Data were collected through a comprehensive literature review of scholarly articles, educational reports, and official documentation on the use of AI tools such as GitHub Copilot, ChatGPT, TensorFlow, and project management platforms like Trello and Jira in teaching contexts. Additionally, observational analysis of case examples where students utilized AI in programming tasks was conducted to illustrate practical applications. The method aims to synthesize insights on how AI supports automation in coding, enhances project management, and facilitates the development of real-world AI models. This approach enables a holistic understanding of AI's pedagogical roles in fostering critical thinking, technical skills, and future readiness among programming students.

RESULTS AND DISCUSSION

General advantages of using digital technologies and artificial intelligence in programming

1. Interactivity: Makes teaching processes interesting and innovative.
2. Personalization: Programs and tools adapted to the abilities of each student.
3. Work efficiency: Saves time for teachers and students through automation.
4. Innovative approach: Teaches advanced technologies to solve complex real-life problems.
5. Easy to master: Simplifies understanding and teaches application of complex concepts.

Teaching programming with artificial intelligence not only improves students' practical skills, but also makes them adaptable to modern technologies. This process increases the efficiency of education and lays the foundation for future technological development.

The results of implementing artificial intelligence (AI) in programming education show a significant increase in student engagement and understanding. Students demonstrated improved problem-solving skills, greater autonomy in learning, and enhanced motivation due to the interactive and adaptive nature of AI-based tools. Discussion of these findings suggests that AI not only facilitates deeper conceptual comprehension through real-time feedback and personalized learning paths but also

prepares students for the demands of a rapidly evolving digital workforce. Moreover, the use of AI streamlines the instructional process for teachers by automating routine tasks, allowing more focus on mentoring and critical thinking development. These outcomes reinforce the notion that integrating AI in programming instruction is not merely a technological upgrade, but a pedagogical advancement that aligns with 21st-century education goals.

CONCLUSION

Fundamental Finding : Artificial intelligence (AI) significantly simplifies learning programming by automating code writing, supporting project management, and enabling students to build AI applications. Tools like GitHub Copilot and ChatGPT help students understand complex algorithms and fix code. Project management platforms such as Trello, Jira, and CI/CD systems enhance collaborative learning. AI also allows students to build real-world applications in machine learning, NLP, computer vision, and deep learning. **Implication :** AI fosters interactivity, personalization, and efficiency in programming education. Students gain practical skills, autonomy, and confidence to solve real-world problems using current technologies. Educators benefit from time-saving automation and enhanced student engagement, making instruction more effective. **Limitation :** Despite its advantages, AI integration may depend heavily on infrastructure, tool accessibility, and students' prior digital literacy. There is limited insight into the long-term impact of AI on critical thinking development and equity across different learning environments. **Future Research :** Further research should explore scalable integration strategies for AI in diverse educational contexts, assess its impact on student creativity and collaboration, and develop inclusive frameworks that ensure all learners benefit from AI-assisted programming instruction.

REFERENCES

Negnevitsky, M. (2005). *Artificial intelligence: A guide to intelligent systems* (2nd ed.). Addison-Wesley.
Goodfellow, I., Bengio, J., & Courville, A. (2016). *Deep learning*. MIT Press.
Webber, J. S. (2021). *Python programming for beginners*. Independently published.
Géron, A. (2019). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems* (2nd ed.). O'Reilly Media.
Martin, R. C. (2008). *Clean code: A handbook of agile software craftsmanship*. Prentice Hall.
Hunt, A., & Thomas, D. (2019). *The pragmatic programmer: Your journey to mastery* (20th Anniversary ed.). Addison-Wesley.
McConnell, S. (2004). *Code complete: A practical handbook of software construction* (2nd ed.). Microsoft Press.
TensorFlow. (n.d.). *TensorFlow documentation*. Retrieved from https://www.tensorflow.org/
PyTorch. (n.d.). *PyTorch documentation*. Retrieved from https://pytorch.org/
Scikit-learn. (n.d.). *Scikit-learn documentation*. Retrieved from https://scikit-learn.org/
GitHub. (n.d.). *GitHub Copilot*. Retrieved from https://github.com/features/copilot

Khudoyberdiev Abdumalik Dilmurodovich
Samarkand Branch of the ISFT Institute, Uzbekistan