Academic Education Publishing House
-AEPH-

Application of Artificial Intelligence in Personalized Teaching: Taking the "Python Programming Design" Course as an Example

Jianmei Chen, Xiaojun Ding*
School of Computer Science and Engineering, Yulin Normal University, Yulin, Guangxi, China
**Corresponding Author.*

Abstract: Taking the course of "Python Programming Design" as an example, this paper explores the personalized teaching of Artificial Intelligence. In the teaching practice, with Student A demonstrating the design of questions, the difficulty is gradually increased according to the learning progress. The case analysis involves the application methods, effect evaluation, feedback from teachers and students, etc. The paper also elaborates on the design practice and effect evaluation methods of Artificial Intelligence teaching in the course. The conclusion points out its progress, problems and optimization strategies. Although the research has limitations, the development trend of Artificial Intelligence teaching is promising. In the future, multiple technologies can be integrated and interdisciplinary teaching can be carried out.

Keywords: Artificial Intelligence; Personalized Teaching; Python Programming Design

1. Introduction

In the era of rapid information technology development, the application of artificial intelligence (AI) in the education field has received increasing attention. The innovation of AI technology has not only transformed teaching methods but also furnished students with a more personalized learning experience [1]. Personalized teaching refers to customized teaching plans and methods based on students' learning needs, interests, and abilities. The traditional teaching model is usually one - size - fits - all and cannot meet the unique requirements of each student. However, AI technology offers a new possibility to achieve the goal of personalized teaching through technical means such as big - data analysis, machine learning, and natural - language processing.

Python programming design is an important course in the field of computer science, covering the basic knowledge and practical skills of programming languages [2]. Thanks to the simplicity, wide application, and abundant development tools of the Python language, it has become the top choice for many beginners and professionals to learn programming [3]. However, traditional Python programming design courses often lack personalization in teaching methods and cannot meet the learning needs and styles of different students [4].

The application of AI in education is not limited to Python programming design. Its potential is equally remarkable in other disciplines as well. For example, in mathematics education, AI can help students better understand mathematical concepts by providing personalized exercise questions and instant feedback. In language education, AI can offer effective language learning tools through natural language processing technology. These applications not only improve learning efficiency but also can better meet the personalized needs of students [5].

In the Python Programming Design course, the application of AI technology mainly focuses on the following aspects:

Adaptive Learning: Through machine learning algorithms, AI can dynamically adjust the teaching content and difficulty according to students' learning progress and performance, thus realizing a personalized learning experience. For example, AI can recommend programming exercises suitable for students based on their programming levels, or provide corresponding help and resources when students encounter difficulties [6].

Intelligent Tutoring: AI can offer intelligent tutoring to help students solve programming problems. For instance, AI can understand students' questions through natural language

processing technology and provide corresponding solutions. In addition, AI can analyze students' code to offer targeted feedback and suggestions, helping students improve their programming skills [7].

Personalized Assessment: AI can conduct a comprehensive assessment of students' learning performance through big data analysis. For example, AI can analyze students' homework completion status, test scores, and classroom participation to evaluate their learning effectiveness and progress. Moreover, AI can also provide personalized learning suggestions and resources according to students' learning styles and preferences [8].

Although the application prospects of AI in education are broad, it also faces some challenges. For example, the complexity and high cost of AI technology may limit its widespread application in education; in addition, the issues of transparency and interpretability of AI systems also need to be addressed to ensure the fairness and impartiality of AI in education [9].

This paper aims to explore specific cases of AI application in personalized teaching. Taking the Python Programming Design course as an example, it shows how to utilize AI technology to achieve personalized teaching. Firstly, we will introduce the basic concepts and principles of AI in personalized teaching. Then, we will describe in detail the specific applications of AI technology in the Python Programming Design course, including aspects such as adaptive learning, intelligent tutoring, and personalized assessment. Finally, we will summarize the advantages and challenges of AI in personalized teaching and propose future research directions.

2. Related Work

When investigating the application of AI in personalized teaching, it is of great importance to understand the existing research and technologies in related fields. This section will provide an overview of the relevant research and technologies that have already applied AI in the education field, especially those applications related to Python programming design. By reviewing these works, we can better understand the potential and challenges of AI in education and offer references for future research and applications.

2.1 Adaptive Learning Systems

Adaptive learning systems are a significant area of AI in education. These systems utilize machine learning algorithms to dynamically adjust teaching content and difficulty according to students' learning progress and performance. In Python programming design courses, adaptive learning systems can recommend suitable programming practice questions for students based on their programming levels, or provide corresponding help and resources when students encounter difficulties [10].

Researchers have explored how to utilize machine learning algorithms to optimize programming education in multiple studies. The research by Wang [11] et al. shows that by analyzing students' code submission records, the learning progress and learning styles of students can be accurately predicted, thereby providing personalized learning suggestions. Additionally, the research by Lin [12] et al. illustrates how to use deep learning algorithms to analyze students' programming problems and provide immediate solutions and feedback.

2.2 Intelligent Tutoring Systems

Intelligent tutoring systems leverage natural language processing technology to assist students in solving programming problems. These systems can understand students' questions and provide corresponding solutions. In Python programming design courses, intelligent tutoring systems can analyze students' code to offer targeted feedback and suggestions, helping students improve their programming skills.

In the research of McNamara [13] et al., it was explored how to utilize natural language processing technology to achieve intelligent tutoring, showing how to understand students' programming problems by means of natural language processing technology and provide corresponding solutions. Additionally, the research of Wang [11] et al. showed how to use deep learning algorithms to analyze students' code and provide targeted feedback and suggestions.

2.3 Personalized Assessment Systems

Personalized assessment systems utilize big data analysis technology to conduct a comprehensive assessment of students' learning performance. These systems can

analyze students' homework completion status, test scores, and classroom participation, thereby evaluating students' learning effectiveness and progress. In Python programming design courses, personalized assessment systems can provide personalized learning suggestions and resources according to students' learning styles and preferences.

In the research of Bagunaid [14] et al., it was explored how to utilize big data analysis technology to achieve personalized assessment, and it was demonstrated how to analyze students' homework completion status and test scores by using big data analysis technology to evaluate students' learning effectiveness and progress. Additionally, the research of Dhananjaya [15] et al. showed how to use deep learning algorithms to analyze students' learning styles and preferences and provide personalized learning suggestions and resources.

In conclusion, the adaptive learning systems, intelligent tutoring systems, and personalized assessment systems have demonstrated great potential in the integrated application of AI in Python programming design courses. They respectively create a higher-quality personalized learning experience for students from different perspectives. Whether by utilizing machine learning algorithms, natural language processing technologies, or big data analysis technologies, they are all striving to conform to the individual differences of students and contribute to the development of education in a more efficient and precise direction.

However, we cannot overlook the existing problems either. From a technical perspective, the development of these systems relies on complex algorithms and models, and their implementation and maintenance costs are relatively high, which poses challenges to the resource investment of educational institutions. Meanwhile, the issues of transparency and interpretability of AI systems remain crucial problems that urgently need to be addressed. These not only relate to the fairness and impartiality of education but also affect the trust and acceptance of these technologies by teachers and students.

Future research directions should focus on how to reduce the technical costs and improve their popularization in the education field. Meanwhile, strengthen the research on the internal logic of AI systems to enhance their transparency and interpretability, enabling teachers and students to better understand and utilize them. In addition, further explore the innovative integration among different technologies. For example, combine the advantages of the three systems more organically to form a more comprehensive and intelligent personalized teaching system, which will bring new breakthroughs and development opportunities to the education field. It is hoped that through continuous research and practice, the application of AI in education can exert its maximum value and promote the educational transformation to move forward in a more positive direction.

3. The Practice of Personalized Teaching in "Python Programming Design"

In the realm of education, the setting of practical programming questions in programming instruction is a vital part in cultivating students' programming capabilities. It needs to be advanced gradually according to students' learning progress and the level of knowledge they have mastered, so as to achieve teaching objectives and enhance teaching effectiveness.

3.1 Practice of Programming Question Setting

In terms of the setting of practice programming questions, the questions initially presented to students are simple output-type questions. For instance, write a Python program to output "Hello, World!" using the print function. This is a classic introductory question for programming learning, aiming to familiarize students with the basic output statements in Python. When a student completes this question and submits it to the AI, the AI will quickly make a judgment. If the answer is correct, the next question might be "Write a program to output your name and age using the print function". If the student answers incorrectly, the AI will not only give the correct answer but also provide a detailed explanation. For instance, print is a built-in function in Python used to output information to the console. In this question, 'Hello, World!' inside the parentheses is a string, which is the content we want to output. After that, the AI will give similar questions, such as "Output 'Python is great!' using the print function", to

test the students again and ensure that they have mastered the basic output method.

As students make progress, the difficulty level of the questions gradually increases. The subsequent questions can be related to variables. For example, "Define a variable named 'name', assign your name to it, and then use the print function to output this variable." If a student makes a mistake at this stage, the AI will point out that "In Python, you use the format of variable_name = value to define a variable. For example, name = 'your name', and then you output the value of the variable through print (name)." Moreover, the AI will present similar questions again, such as "Define a variable named 'age', assign your age to it, and then output this variable", until the student fully masters the methods of defining and using variables.

After that, the questions can involve simple mathematical operations. For example, "Define two variables, num1 and num2, and assign them the values of 5 and 3 respectively, and then use the print function to output their sum." If a student makes a mistake, the AI will explain the operation rules of variables and how to correctly output the operation results in the print function. Subsequent questions of a similar nature can be about outputting the difference, product, quotient, etc. of the two variables, gradually deepening students' understanding and application ability of basic operations in programming.

When students have a relatively solid grasp of the above basic knowledge, the learning of branching statements will be introduced. First, a simple branching statement question is presented: Write a program where an integer is input and use the if statement to determine whether this number is greater than 10. If it is, output "This number is greater than 10". In this question, students need to understand that the input () function is used to obtain user input, and they should also know how to convert the input value to the integer type (using the int () function), as well as the basic structure of the if statement. If a student answers incorrectly, the AI will give the correct Python code as follows:

```
num = int(input("Please enter an integer: "))
if num > 10:
    print("This number is greater than 10")
```

At the same time, a detailed explanation will be provided: Firstly, the input () function is used to obtain the string entered by the user. Secondly, the int () function is used to convert this string into an integer and store it in the variable num. Afterwards, the if statement is used to determine whether num is greater than 10. If the condition is met, the print statement will be executed to output the corresponding content. Subsequently, the AI will present similar questions, such as determining whether the input number is less than 5.

Here is a more complex branching statement question: Write a program to input an integer and use the if-else statement to determine whether the number is positive, negative, or zero, and output the corresponding information (if the number is positive, output "This is a positive number"; if the number is negative, output "This is a negative number"; if the number is zero, output "This is zero"). The correct Python code is as follows:

```
num = int(input("Please enter an integer: "))
if num > 0:
    print("This is a positive number")
elif num < 0:
    print("This is a negative number")
else:
    print("This is zero")
```

When explaining, the AI will highlight the multi-layer nested structure of the if-else statement as well as the logical judgment relationship. Similar questions could be to output different information based on the size relationship of the two input numbers. For instance, input two numbers and judge whether the first number is greater than the second number. If it is, output "The first number is greater than the second number". If they are equal, output "The two numbers are equal". Otherwise, output "The first number is less than the second number".

Only after students have fully grasped the branching statements will the difficulty of the questions be increased. For example, starting from simple for statement questions and then moving on to more complex for statement questions. In this way, through the step-by-step and easy-to-difficult question design, the gradual accumulation of knowledge and the steady improvement of abilities in personalized teaching can be achieved.

3.2 Improvements in the Setting of Practical Programming Questions

To further enhance the application effect of AI

in the personalized teaching of Python programming design, we have made the following improvements:

First of all, strengthen interaction and feedback. Incorporating more elements of interaction and feedback into the question and explanation sections will help students gain a deeper understanding and better mastery of the knowledge. For example, an interactive part can be set up after each question to encourage students to independently solve similar problems or submit their code, and the AI will provide more comprehensive feedback and improvement suggestions.

Secondly, increase comprehensive application questions. Introducing such questions in each learning stage can prompt students to apply the knowledge they have learned to solve practical problems. For example, after learning branching statements and loop statements, comprehensive application questions can be assigned, asking students to write programs to solve practical problems.

Furthermore, strengthen evaluation and feedback. Add an evaluation part in each learning stage, asking students to submit their code. Based on this, the AI can provide more feedback and improvement suggestions, thus enabling students to better master the knowledge.

In addition, enrich resources and references. Equip each learning stage with more resources, such as tutorials, videos, articles, etc., to assist students in understanding and mastering the learned content.

Finally, highlight personalization. Dynamically adjust the teaching content and difficulty according to the students' learning progress and performance, and provide more personalized learning suggestions and resources in each learning stage.

Through the implementation of these improvement measures, the application of AI in the personalized teaching of Python programming design can be more complete and practical, which greatly helps students in understanding and mastering knowledge, creates a higher - quality and more efficient learning experience for students, and promotes personalized teaching to a new height.

4. Conclusion and Prospect

AI technology holds strong potential for educational applications. For instance, it can dynamically adjust teaching content and difficulty based on students' learning progress and performance to achieve a personalized learning experience. It can provide instant feedback and suggestions to help students master knowledge more efficiently. It is also capable of recommending suitable learning resources according to students' learning needs and interests to enhance learning effectiveness. Meanwhile, through machine learning algorithms, natural language processing technology, and big data analysis, it can respectively complete the adjustment of teaching content, the understanding and answering of students' questions, and the comprehensive assessment of learning performance.

However, it also faces numerous problems. The complexity and high cost of AI technology itself may constrain its wide application in the education field. Its decision-making process is often like a "black box" operation, which is difficult for humans to understand and explain. Moreover, when dealing with a large amount of student data, ensuring the privacy and security of the data is a major challenge. In addition, algorithmic biases may also have a negative impact on students' learning experience and effectiveness.

Although the application of AI in personalized teaching faces challenges, its potential in the education field remains huge. Future research and application can be envisioned from the following aspects: First, improve the transparency and interpretability of AI systems. Develop new algorithms and methods to make the decision - making process more transparent and interpretable, enhance the trust of educators and students, and at the same time use visualization techniques to help them better understand the decision - making process. Second, reduce the implementation cost of AI systems. Develop more efficient algorithms and utilize open - source technologies to make the application of AI in education more extensive. Third, explore new applications of AI technology. For example, use blockchain technology to ensure the security and privacy of educational data, use virtual reality technology to provide an immersive learning experience, and use emotion analysis technology to understand students' emotional states and provide

personalized emotional support.

Through continuous exploration and application of AI technology, we can further improve the quality of education, help students master knowledge, and also provide more tools and resources for educators. It is hoped that this article can provide practical suggestions for educators to help them better apply AI technology and improve students' learning effectiveness and experience.

References

[1].Van der Vorst T and N Jelicic. Artificial Intelligence in Education: Can AI bring the full potential of personalized learning to education? 30th European Conference of the International Telecommunications Society: "Towards a Connected and Automated Society". 2019.

[2].Brusilovsky P, Malmi L, Hosseini R, et al. An integrated practice system for learning programming in Python: design and evaluation. Research and practice in technology enhanced learning, 2018. 13: p.1-40.

[3].Orr JW and N Russell. Automatic assessment of the design quality of python programs with personalized feedback. arXiv preprint arXiv:2106.01399, 2021.p.1-8.

[4].Holsapple K and AC Bart. Designing Designer: The Evidence-Oriented Design Process of a Pedagogical Interactive Graphics Python Library. in Proceedings of the 53rd ACM Technical Symposium on Computer Science Education, 2022.1: p. 85-91.

[5].Hashim S, Omar MK, Ab Jalil H, et al. Trends on technologies and artificial intelligence in education for personalized learning: systematic literature. Journal of Academic Research in Progressive Education and Development, 2022. 12(1): p. 884-903.

[6].Bhutoria A. Personalized education and artificial intelligence in the United States, China, and India: A systematic review using a human-in-the-loop model. Computers and Education: Artificial Intelligence, 2022. 3: p. 100068.

[7].Al-Badi A and A Khan. Perceptions of learners and instructors towards artificial intelligence in personalized learning. Procedia computer science, 2022. 201: p. 445-451.

[8].Rivers C and A Holland. Management education and artificial intelligence: Toward personalized learning, in The future of management education. Routledge, 2022. p. 184-204.

[9].Pratama MP, R Sampelolo and H Lura. Revolutionizing education: harnessing the power of artificial intelligence for personalized learning. Klasikal: Journal of education, language teaching and science, 2023. 5(2): p. 350-357.

[10].Kazemitabaar M, Chyhir V, Weintrop D, et al. Codestruct: Design and evaluation of an intermediary programming environment for novices to transition from scratch to python. in Proceedings of the 21st Annual ACM Interaction Design and Children Conference, 2022. p. 261-73

[11].Wang L, Sy A, Liu L, et al. Deep knowledge tracing on programming exercises. in Proceedings of the fourth (2017) ACM conference on learning@ scale, 2017. p. 201-04.

[12].Lin P and S Chen. Design and evaluation of a deep learning recommendation based augmented reality system for teaching programming and computational thinking. IEEE Access, 2020. 8: p. 45689-45699.

[13].McNamara DS, SA Crossley and R Roscoe. Natural language processing in an intelligent writing strategy tutoring system. Behavior research methods, 2013. 45: p. 499-515.

[14].Bagunaid W, N Chilamkurti and P Veeraraghavan. Aisar: Artificial intelligence-based student assessment and recommendation system for e-learning in big data. Sustainability, 2022. 14(17): p. 10551.

[15].Dhananjaya GM, Goudar RH, Kulkarni A, et al. A Digital Recommendation System for Personalized Learning to Enhance Online Education: A Review. IEEE Access, 2024. 12. p. 34019–34041