engineering proceedings

MDPI

Proceeding Paper

Application of AI-Driven Personalized Training Platform for Operating System Courses †

Xiaohan Long, Jing Zhang and Xingzhi Chang *

School of Software and Big Data, Changzhou College of Information Technology, Changzhou 213164, China; longxiaohan@czcit.edu.cn (X.L.); zhangjing@czcit.edu.cn (J.Z.)

* Correspondence: changxingzhi@czcit.edu.cn

† Presented at the 9th Eurasian Conference on Educational Innovation 2026 (ECEI 2026), Da Nang City, Vietnam, 30 January–2 February 2026.

Abstract

Challenges in operating system courses, such as inefficient evaluation of practical skills, delays in debugging, and a lack of personalized instruction, were addressed by an AI-driven platform developed in this study. The platform enables precise teaching and efficient debugging through real-time compliance verification, intelligent error diagnosis, and dynamic generation of personalized tasks. By utilizing full-process behavioral data for multidimensional competency assessment, the platform was deployed in over 20 classes. It effectively reduced instructors' workload while enhancing students' theoretical knowledge and practical understanding, promoting efficient, personalized, and intelligent transformation in operating system training.

Keywords: vocational education modernization; AI-driven education; personalized learning; intelligent evaluation

1. Introduction

Operating System (OS) courses play a critical role in computer science education by fostering systematic thinking, practical skills, and engineering literacy. However, traditional teaching methods, reliant on theoretical instruction and limited hands-on exercises, struggle to meet modern demands for personalized, practical, and innovative learning. Key challenges include inefficient skill assessment, delayed error debugging, uniform content that ignores individual differences, and oversimplified evaluation metrics [1].

Advancements in AI offer promising solutions. AI applications in education have evolved from theory to practice, demonstrating significant progress in personalized learning, intelligent tutoring, and analytics [2]. Structured generative AI enhances higher-order thinking and programming logic [3], while AI-supported flipped classrooms improve engagement and outcomes [4]. This study develops an AI-driven platform for OS courses, integrating constructivist, adaptive, and personalized learning theories [5] to address core teaching limitations.

On the basis of the technology acceptance model, the platform was used to analyze perceived usefulness and ease of use. The results show that personality traits, including openness and conscientiousness, influence students' acceptance of generative AI, informing the platform's design [6]. Theoretically, distributed creativity, collaborative creativity, and scaffolding theories are combined into a unified framework for personalized technical education, bridging pedagogy and technology.

Academic Editors: Teen-Hang Meen, Chun-Yen Chang and Cheng-Fu Yang

Published: 3 June 2026

2. Background Knowledge

OS course reform remains a priority in computer education, yet traditional approaches often feature outdated content, passive learning, and limited experimentation. While an interactive system for CPU scheduling concepts reduces cognitive load through simulation-real-time integration, it focuses narrowly on scheduling algorithms. In personalized learning, studies report positive outcomes in secondary and higher education [7], with methods such as AI-driven recommendations and adaptive analytics. A large language model-based recommendation system boosts learning effectiveness [8], especially for beginners, and a learning-style framework enables scalable personalization [9].

Human–computer interaction research highlights AI's role in metacognitive training using assessment-feedback-reflection-practice cycles [10]. Additionally, randomized experiments reveal that early AI exposure enhances creative ideation [11]. However, existing AI education research has notable gaps: previous studies focus on theoretical or general skills, with limited tools for technical courses such as OS; personalized systems prioritize content over practical support; assessments overemphasize grades while neglecting comprehensive skills; and most lack large-scale empirical validation.

The study results offer an integrated approach to innovation across three key levels. At the technical level, a comprehensive AI system is designed to bridge the gap between theoretical knowledge and practical skill development. In practice, its effectiveness is supported by large-scale evidence gathered from technical courses. From an educational perspective, the developed platform balances learning outcomes with equity by optimizing resource allocation. The results of this study serve as a reference for the ongoing transformation of educational models and technology.

3. System Design

3.1. System Architecture

The system adopts a layered architecture design, with clear responsibilities for each layer and loose coupling between modules to facilitate maintenance and function expansion. The real-time principle ensures timely response to key business operations, providing students with instant feedback. Security design covers identity authentication, data protection, and access control to ensure the security and privacy of teaching data. The architecture is shown in Figure 1.

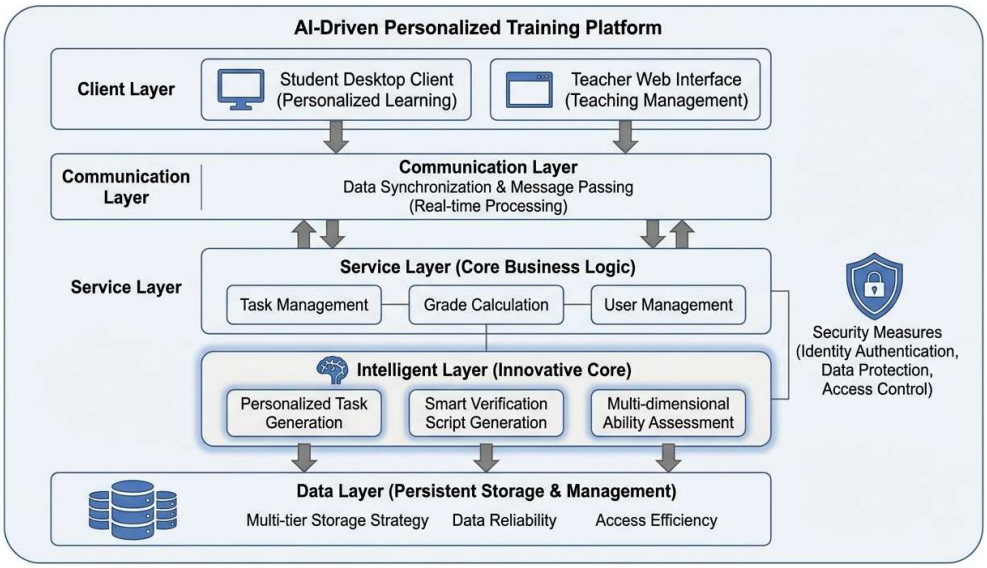


Figure 1. System architecture.

https://doi.org/10.3390/engproc2026141003

3.2. Core Module Design

3.2.1. Personalized Task Generation Module

Leveraging student competency profiles and behavioral data, this module dynamically generates personalized learning tasks appropriate to current skill levels by analyzing multiple dimensions, including operational accuracy and command standardization. It customizes task parameters and descriptions according to individual learning preferences while automatically adjusting difficulty to maintain optimal challenge levels. The system also provides completion time estimates to assist teachers in instructional planning.

3.2.2. Intelligent Verification Script Module

This module is responsible for auto-generating lightweight verification code, utilizing a template library of standardized verification patterns to enable real-time operation checking across different task types. The generated code undergoes optimization for a minimal memory footprint and fast execution while incorporating security checks to prevent malicious code execution. Performance validation ensures all verification scripts meet system requirements before deployment.

3.2.3. Multidimensional Competency Assessment Module

This evaluation module analyzes full-process behavioral data across five dimensions using a weighted scoring algorithm to assess operational accuracy, command standardization, and problem-solving capabilities. It tracks learning progress through trend analysis while generating visual assessment reports and personalized recommendations. The system provides targeted improvement suggestions based on evaluation outcomes to guide student development.

3.3. Data Flow Design

Employing an event-driven architecture with real-time processing pipelines, the system manages high-concurrency data flow through a complete collection–transmission–processing–storage–analysis workflow that ensures data integrity and traceability. Three specialized data stream types handle distinct responsibilities: real-time flows process immediate operations such as task verification, batch flows manage analytical tasks including competency assessment, and feedback flows enable closed-loop control by returning processed results to the user interface. The process is shown in Figure 2.

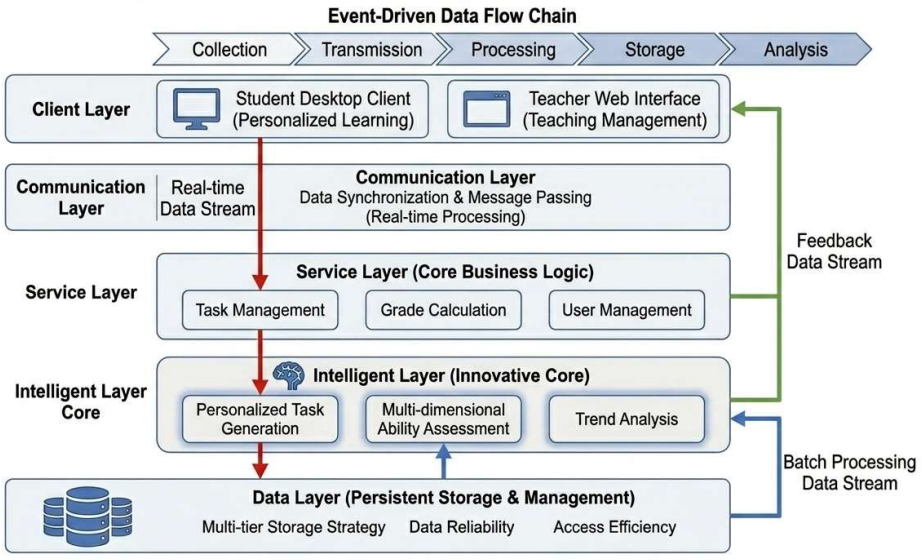


Figure 2. Data flow architecture.

3.4. Hardware Component

Technologies for this system were selected according to functional and non-functional requirements. For the front end, the PySide6 framework (version 6.9.1) was chosen due to its cross-platform compatibility, high performance, and extensive UI component library. The back end utilizes the Flask framework (version 3.1.1), which provides a lightweight and scalable foundation for building robust representational state transfer application programming interface services. Data was managed by a combination of My Structured Query Language (MySQL) and Redis. MySQL (version 8.0.4) offers stable transaction support and complex query capabilities for structured data, while Redis (version 7.1.0) provides high-performance, in-memory operations to meet the low-latency demands of real-time processing. To ensure these data insights are accessible, the Chart.js library (version 4.5.0) is used to generate intuitive visualizations for teaching analysis. The system's core functionality is divided into three specialized modules.

- Personalized task generation module uses machine learning algorithms to build detailed student competency profiles;
- Intelligent verification script module employs template engines and code generation technologies for automated assessment;
- Multidimensional competency assessment module manages comprehensive data collection and analysis throughout the entire learning process.

4. Platform Data Analysis

The AI-driven personalized training platform for OS courses has been deployed in over 20 classes, covering more than 1000 student users. During system operation, it has accumulated over 3,000,000 student behavior records, providing a rich data foundation for personalized teaching analysis. The platform adopts a distributed deployment architecture, with servers deployed in a cloud server cluster and clients connected to the teaching environment via a local area network. The system's average response time is controlled within 2 s, and the real-time feedback latency for task checks is less than 500 milliseconds, meeting the real-time requirements of teaching scenarios.

4.1. Platform Operation Effect

The teacher-side task release interface is shown in Figure 3, which supports quick task creation and batch release. The system provides a rich task template library, allowing teachers to select preset templates or create custom tasks according to teaching objectives.

Figure 3. Teacher-side task release.

The task release process implements intelligent configuration, with the system automatically recommending appropriate check items and scoring standards based on task types. Released tasks are notified to the student side via a real-time push mechanism, ensuring the timely delivery of teaching instructions. The student-side task execution interface is shown in Figure 4, which provides clear task guidance and real-time operation feedback. The system conducts real-time checks on students' operations through intelligent verification scripts and automatically generates detailed scoring reports.

Figure 4. Student-side task automatic scoring.

The automatic scoring mechanism implements objective and fair evaluation standards, avoiding subjective biases in manual scoring. The scoring report not only includes the total score but also details the scoring results of each checkpoint, helping students accurately identify problems in their operations. Based on students' learning behavior data, the system dynamically adjusts task difficulty and content. Students with weaker competencies receive more basic training tasks, while students with stronger competencies obtain more challenging advanced tasks. The personalized recommendation engine pushes suitable learning resources and practice tasks according to students' knowledge mastery and learning preferences. The learning path planning function provides students with optimal learning sequence suggestions to improve learning efficiency.

The teacher-side score management interface is shown in Figure 5. The platform provides multi-dimensional score statistical analysis functions. The system automatically generates class score distribution charts, personal competency radar charts, and progress trend analyses, helping teachers fully grasp students' learning status. The score export function supports output in multiple formats, facilitating teachers in conducting further data analysis and teaching research. The intelligent early warning system identifies students with learning difficulties and promptly reminds teachers to provide targeted tutoring.

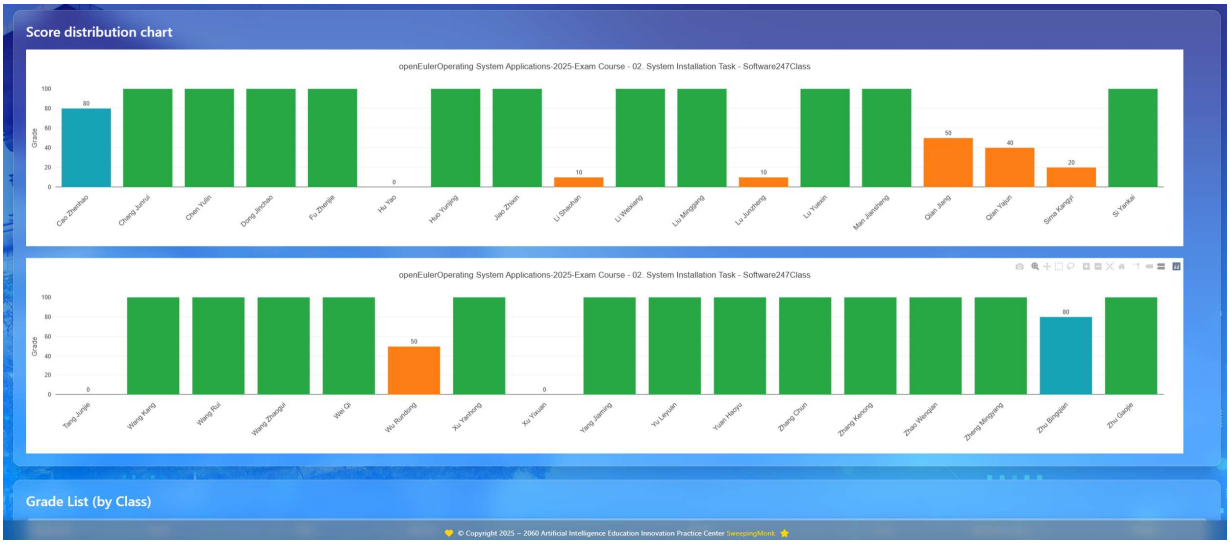
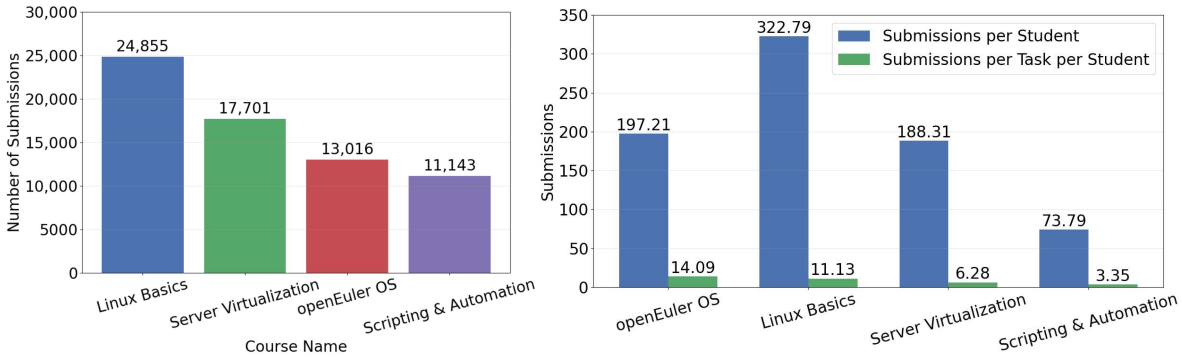


Figure 5. Teacher-side score statistics.

4.2. Platform Data Statistics

The teaching platform has successfully supported the teaching practice of multiple IT courses, including core courses such as Linux Basics, openEuler OS, Server Virtualization Technology, and Script Development & Automated Operations. The platform adopts a task-driven teaching model, with task settings for each course as follows: 29 tasks were released for the Linux Basics course, 14 for the openEuler OS course, 30 for the Server Virtualization Technology course, and 22 for the Script Development & Automated Operations course. The total number of task submissions for each course is shown in Figure 6a. Figure 6b shows the average number of submissions per student. From the data perspective, although the Linux Basics course leads in absolute submission numbers, the openEuler OS course performs most prominently in the indicator of average number of submissions per student per task, with an average of 14.09 submissions per student per task. This data indicates that students in the openEuler course invested more attempts and practice in each task. The Linux Basics course follows closely, with an average of 11.28 submissions per student per task, reflecting a learning model where students master basic operational skills through repeated practice.



(**a**) Course Submission Frequency (**b**) Comparison of Submission Metrics by Course

Figure 6. Number of task submissions.

The platform supports a learning mechanism where students can submit tasks multiple times for iterative improvement. This design is fully reflected in the performance results. Figure 7 shows the average submission score per course. From the average of students'

highest scores in each course, the Linux Basics course achieved an excellent average score of 97.5 points, indicating that through the platform's repeated practice mechanism, students have basically mastered the core operational skills of the course. The Script Development and Automated Operations course also performed well, with an average highest score of 96.42 points.

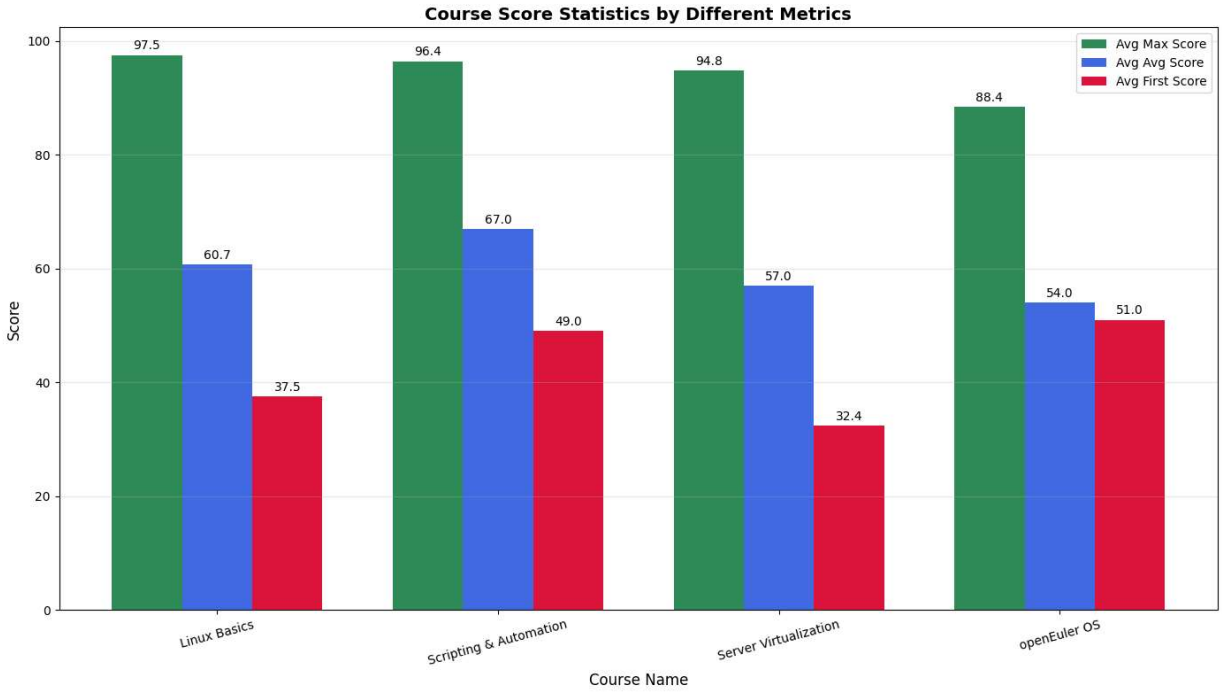


Figure 7. Average submission score per course.

From the comparative analysis of the first submission score and the final score, the average first submission score of the Linux Basics course was only 37.53 points, while the final average score increased to 97.5 points after multiple attempts, representing an increase of up to 60 points. This fully proves the effectiveness of the platform's iterative learning model. Similarly, the Server Virtualization Technology course increased from an initial submission score of 32.42 points to 94.75 points, with an increase of over 62 points, demonstrating significant progress in students' learning process.

To systematically evaluate the application effect of this teaching platform, this study adopts a rigorous controlled experiment design. Two parallel classes with identical teaching content, instructors, and evaluation standards were selected, designated as the experimental group (n = 30) and the control group (n = 28). The experimental group used this teaching platform for auxiliary teaching throughout the course, while the control group maintained the traditional teaching model. After the course, a comparative analysis was conducted on the comprehensive scores of the two groups of students, with the total score evaluation based on a unified multi-dimensional evaluation standard covering knowledge mastery, skill operation, and professional literacy.

As shown in Figure 8, the average score of the experimental group was 83.83 points (standard deviation = 7.42), slightly higher than that of the control group (82.25 points, standard deviation = 10.71). From the perspective of data distribution characteristics, the score dispersion of the experimental group was significantly smaller than that of the control group, with a standard deviation reduction of 30.8% compared with the control group. This indicates that the application of the platform helps stabilize students' learning performance and reduce score fluctuations. Notably, the control group had a wide score range (47–95 points) and extremely low scores, while the experimental group had a relatively

concentrated score distribution (65–97 points) and more balanced overall performance. It can be concluded that this teaching platform demonstrates a positive effect in improving learning stability, providing valuable references for subsequent teaching reforms.

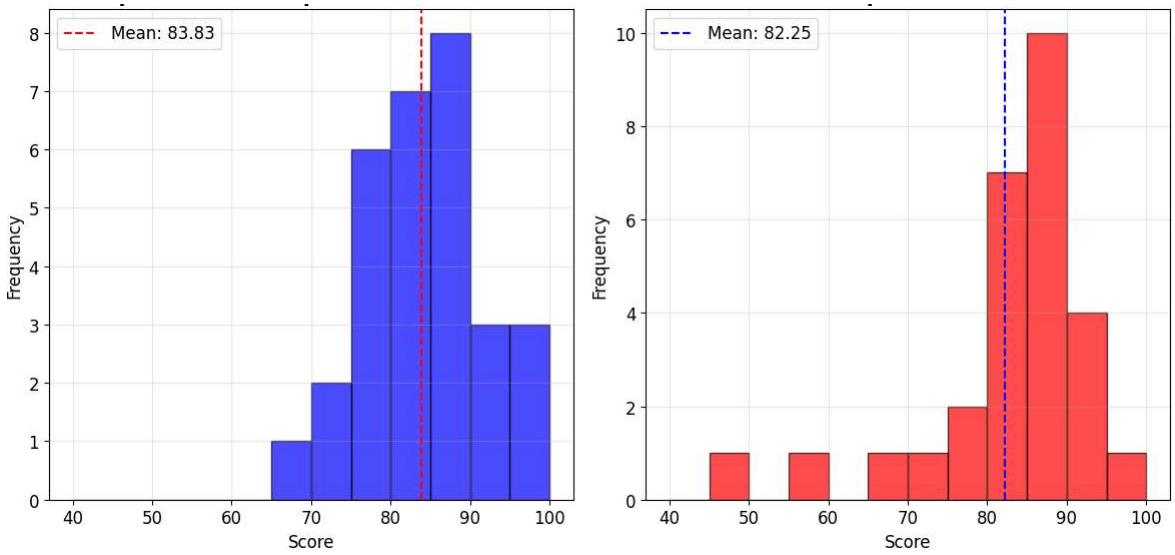


Figure 8. Score distribution.

5. Conclusions

We implemented and verified an AI-driven personalized training platform for OS courses, providing innovative solutions to the challenges faced by traditional OS course teaching. Research results showed that the platform exhibits significant advantages in enhancing teaching efficiency, facilitating personalized learning, and improving learning effects.

Author Contributions: Methodology, X.L.; software, X.L.; writing—original draft preparation, X.L.; investigation, J.Z.; resources, J.Z.; validation, J.Z.; project administration, X.C.; funding acquisition, X.C.; writing—review and editing, X.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the 2025 Research Project of the Chinese Vocational and Technical Education Association: "Exploration of the SPOC Teaching Model Based on the Trinity Integration of Resources, Platform, and AI" (SZ25A42); the 2025 Teaching Special Project of the National Association of Computer Education in Universities: "Research and Practice of Digital Teaching Reform in Vocational Education Courses Empowered by Information Technology" (2025-AFCEC-018); the 2024 Information Technology Curriculum Teaching Reform Research Project of National Higher Vocational Colleges: "Research and Practice of Data-Driven Smart Teaching Model" (KT2024053); the 2023 Provincial Education Science Planning Project: "Research on the Construction of Learning Support System for Self-Taught Examinations" (K-A/2023/09); and the Jiangsu Province Qinglan Project.

Institutional Review Board Statement: Ethical review and approval were waived for this study because the research involved routine educational practices and did not pose any additional risks to participants. The study was conducted as part of normal teaching activities, and all data were anonymized and analyzed retrospectively.

Informed Consent Statement: Informed consent was waived because the study involved routine educational practices with anonymized data and posed no more than minimal risk to participants. All data were analyzed retrospectively.

Data Availability Statement: The data that support the findings of this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Chen, X. Practical research on mixed teaching reform of Linux operating system course. *Contemp. Educ. Teach. Res.* **2024**, *5*, 1–6. [CrossRef]
2. Alzubaidi, K. The role of generative AI in higher education: Institutional guidelines, generational gaps, and the Grok 4 challenge. *Arab World Engl. J.* **2025**, *11*, 1–4. [CrossRef]
3. Nathaniel, J.; Oyelere, S.S.; Suhonen, J.; Tedre, M. Investigating the impact of generative AI integration on the sustenance of higher-order thinking skills and understanding of programming logic. *Comput. Educ. Artif. Intell.* **2025**, *9*, 100460. [CrossRef]
4. Nadeem, P.K.; Kumar, B.H.; Sonar, R.M.; Lamba, A.K.; Valiveti, V.S.M.; Kalra, G. Enhancing student engagement and learning outcomes in higher education through e-learning and restricted Boltzmann machines. In Proceedings of the 2025 3rd International Conference on Data Science and Information System (ICDSIS), Hassan, India, 16–17 May 2025.
5. Li, J. Evaluation of entrepreneurship education course effect and personalized learning path using CART decision tree. *Int. J. Knowl. Manag.* **2024**, *21*, 1–22. [CrossRef]
6. Arpaci, I.; Kuci, I.; Gibreel, O. The role of personality traits in predicting educational use of generative AI in higher education. *Sci. Rep.* **2025**, *15*, 30440. [CrossRef] [PubMed]
7. Yang, Q.; Liang, C. A second-classroom personalized learning path recommendation system based on large language model technology. *Appl. Sci.* **2025**, *15*, 7655. [CrossRef]
8. Tudor, I.; Holenko Dlab, M.; Hoić-Božić, N. Personalized learning in secondary and higher education: A systematic literature review of technology-enhanced approaches. *Int. J. Educ. Methodol.* **2025**, *11*, 359–375. [CrossRef]
9. Fletscher, L.; Mercado, J.; Gómez, A.; Mendoza-Cardenas, C. Innovating personalized learning in virtual education through AI. *Multimodal Technol. Interact.* **2025**, *9*, 69. [CrossRef]
10. Li, W.; Cui, X.; Manoharan, P.; Dai, L.; Liu, K.; Huang, L. AI-assisted feedback and reflection in vocal music training: Effects on metacognition and singing performance. *Front. Psychol.* **2025**, *16*, 1598867. [CrossRef] [PubMed]
11. Baltà-Salvador, R.; Brasó-Vives, E.; Pea, M. Evaluating AI-assisted creative ideation: A crossover study in higher education. *Think. Ski. Creat.* **2025**, *59*, 101958. [CrossRef]