Quest Journals
Journal of Electronics and Communication Engineering Research
Volume 12 ~ Issue 1 (Jan.-Feb. 2026) pp: 01-08
ISSN(Online): 2321-5941
www.questjournals.org

Research Paper

Enhancing Computer Teaching Using AI-Based Personalised Learning Tools

Aremu Abdulsalam and Ajiwe Olabanji Peter
Department of Computer Science
Nigerian Army School of Education, Ilorin, Kwara State, Nigeria

Abstract
The integration of artificial intelligence (AI) into computer science education represents a transformative opportunity to address persistent challenges of learner diversity, instructional personalization, and scalable quality education. This study investigates the development, implementation, and effectiveness of AI-based personalized learning tools in enhancing computer teaching within resource-limited educational settings in Nigeria. Grounded in constructivist learning theory, adaptive learning principles, and self-regulated learning frameworks, we developed a comprehensive AI-powered learning platform incorporating intelligent tutoring systems, adaptive content delivery, automated assessment with immediate feedback, learning analytics dashboards, and personalized learning path recommendations. Employing a mixed-methods research design, we conducted a quasi-experimental study involving 300 senior secondary school students across eight schools in Kwara State, comparing learning outcomes between students using the AI-based platform (experimental group, n=150) and those receiving conventional computer science instruction (control group, n=150) over a 12-week period. Quantitative results demonstrated substantial improvements across multiple outcome dimensions: programming skills increased by 73% (Cohen's d = 2.14), conceptual understanding of computer science principles improved by 61% (Cohen's d = 1.87), problem-solving abilities enhanced by 58% (Cohen's d = 1.65), and student engagement and self-efficacy increased by 67% (Cohen's d = 1.92), all statistically significant at p < 0.001. Learning analytics revealed that AI-driven personalization enabled students to progress at individualized paces while maintaining high mastery levels, with the system successfully identifying and addressing knowledge gaps in real-time. Qualitative analysis of teacher perspectives (n=32) identified benefits including reduced instructional workload, enhanced differentiation capabilities, improved student motivation, and valuable data-driven insights, alongside implementation challenges such as initial setup complexity, technology infrastructure requirements, and the need for professional development in AI-mediated pedagogy. Cost-effectiveness analysis demonstrated that despite initial development investments, the platform achieved 60% cost reduction per student compared to traditional computer lab infrastructure over a three-year period when deployed at scale. This research contributes theoretical insights into AI's role in personalized learning, validated design principles for educational AI systems in resource-constrained contexts, and empirical evidence of effectiveness across cognitive, affective, and behavioral learning dimensions. Findings have significant implications for computer science education policy, teacher preparation programs, and strategies for leveraging emerging technologies to achieve Sustainable Development Goal 4 (quality education) in developing countries.

Keywords: *Artificial intelligence, personalized learning, adaptive learning systems, computer science education, intelligent tutoring, learning analytics, educational technology, Nigeria*

*Received 12 Jan., 2026; **R**evised 23 Jan., 2026; **A**ccepted 25 Jan., 2026 © The author(s) 2026.*
P**ublished with open access at **www.questjournas.org

I. Introduction

Computer science education has become essential for preparing students to participate effectively in increasingly digital economies and societies. As technological advancement accelerates and digital transformation permeates all sectors, computational thinking, programming skills, and digital literacy represent fundamental competencies comparable to traditional literacy and numeracy (Wing, 2006). Recognizing this imperative, educational systems globally have expanded computer science curricula, with many countries mandating computer education from primary through secondary levels (Grover & Pea, 2013). However, effective computer science education faces significant pedagogical challenges stemming from learner diversity, the abstract nature of computational concepts, rapid technological change, and substantial resource requirements.

1.1 The Challenge of Learner Diversity in Computer Science Education

Computer science classrooms are characterized by exceptional learner diversity across multiple dimensions that complicate instruction. Students enter computer courses with vastly different prior technology exposure, ranging from those with extensive programming experience to complete novices who have never written code (Guzdial, 2015). This experiential diversity creates instructional dilemmas: teaching to novices may bore experienced students, while teaching to advanced learners may overwhelm beginners.

Beyond experiential differences, students exhibit diverse cognitive styles, learning preferences, and aptitudes for abstract thinking and logical reasoning central to programming (Robins et al., 2003). Research in computing education identifies distinct learner profiles—some students excel at systematic, step-by-step problem decomposition while others thrive in exploratory, trial-and-error environments; some prefer visual representations while others favor textual code; some learn best through concrete examples while others grasp abstract principles readily (Pashler et al., 2008). Traditional one-size-fits-all instruction struggles to accommodate this cognitive diversity effectively.

Furthermore, students progress at different paces. In programming education particularly, understanding foundational concepts like variables, loops, and functions is prerequisite for more advanced topics like object-oriented programming and data structures. Students who fail to master prerequisites experience compounding difficulties, while those who master concepts quickly become disengaged waiting for peers (Soloway & Spohrer, 1989). This pace variation challenges lockstep curriculum delivery.

In Nigerian secondary schools, these diversity challenges are compounded by resource constraints. Computer science teachers often manage classes exceeding 40 students with limited computers, making individualized attention practically impossible (Okebukola, 2019). Teachers resort to lecture-demonstration methods where students watch programming demonstrations on a single screen rather than engaging in hands-on practice, fundamentally undermining the experiential nature of learning to program (Olibie & Akudolu, 2013). Assessment typically involves paper-based examinations testing theoretical knowledge rather than authentic programming skills, creating misalignment between learning objectives and evaluation methods.

1.2 The Promise of AI-Based Personalised Learning

Artificial intelligence technologies offer transformative potential for addressing learner diversity through personalized, adaptive learning systems that adjust instruction to individual student needs, abilities, and progress (Luckin et al., 2016). AI-based personalized learning leverages machine learning algorithms, natural language processing, and intelligent tutoring architectures to create educational experiences tailored to each learner's unique profile, preferences, and performance trajectory (Woolf, 2010).

Key capabilities of AI-based personalized learning systems include:

- Adaptive Content Delivery: AI algorithms analyze student performance patterns to determine optimal sequencing, pacing, and presentation of learning materials. Content adapts dynamically—students mastering concepts quickly receive advanced challenges while those struggling receive additional scaffolding, remediation, and alternative explanations (VanLehn, 2011).
- Intelligent Tutoring: AI tutors provide individualized guidance, hints, and feedback mimicking expert human tutors. These systems recognize common misconceptions, identify knowledge gaps, and offer targeted interventions addressing specific learning needs (Kulik & Fletcher, 2016).
- Immediate Formative Feedback: Automated assessment systems evaluate student work instantly, providing detailed feedback on errors, suggestions for improvement, and opportunities for revision. Immediate feedback enables rapid error correction and iterative improvement impossible with delayed teacher assessment (Shute, 2008).
- Learning Analytics: AI systems collect and analyze granular data on student interactions, performance, time allocation, and progress. Analytics dashboards provide teachers with insights into individual and class-level learning patterns, enabling data-driven instructional decisions and early intervention for struggling students (Siemens & Baker, 2012).
- Personalized Learning Paths: Based on assessment results, learning preferences, and goals, AI systems recommend individualized learning paths guiding students through optimal sequences of content, activities, and assessments (Bulger, 2016).
- Scalable One-to-One Tutoring: AI systems provide personalized support to every student simultaneously, achieving individualization impossible for human teachers managing large classes. This scalability is particularly valuable in resource-limited settings where teacher-student ratios are high (Bloom, 1984).

In computer science education specifically, AI-based personalization addresses unique challenges. Programming involves both conceptual understanding and procedural skill—students must grasp abstract computational concepts while developing practical coding abilities. AI systems can scaffold both dimensions, providing conceptual explanations while simultaneously offering hands-on programming practice with intelligent feedback on code quality, efficiency, and correctness (Rivers & Koedinger, 2017). Automated code analysis detects syntax errors, logical bugs, and style issues, offering specific guidance for improvement. This immediate, detailed feedback accelerates skill development compared to delayed teacher review (Narciss & Huth, 2004).

Moreover, AI systems gamify learning through progress tracking, achievement badges, and adaptive challenges maintaining optimal difficulty—neither too easy (causing boredom) nor too hard (causing frustration). This zone of proximal development targeting enhances motivation and engagement, addressing persistent challenge of student disinterest in computing (Deterding et al., 2011).

1.3 The Nigerian Context: Opportunities and Challenges

Nigeria's context presents both compelling rationale and unique challenges for AI-based personalized learning in computer science education. As Africa's most populous nation with over 200 million people and median age under 18, Nigeria has an enormous youth population requiring quality education for economic development (World Bank, 2021). The Nigerian government recognizes computer science education's strategic importance, mandating computer studies in secondary schools and emphasizing science, technology, engineering, and mathematics (STEM) education in national policy (Federal Ministry of Education, 2020).

However, significant challenges constrain effective computer science education:

1. **Infrastructure Deficiencies:** Most Nigerian secondary schools lack adequate computer laboratories. A national survey found only 23% of schools have functional computer labs meeting minimum standards, with rural schools particularly disadvantaged (Federal Ministry of Education, 2018). Even schools with labs often have outdated equipment, unreliable electricity, limited internet connectivity, and insufficient computers for student-to-computer ratios enabling hands-on practice.
2. **Teacher Capacity Gaps:** Many computer science teachers lack adequate preparation in both subject matter and pedagogy. Some teachers assigned to computer science hold qualifications in unrelated disciplines and have minimal programming expertise themselves (Ofozoba & Ofozoba, 2024). Professional development opportunities are limited, leaving teachers unable to provide effective instruction or stay current with rapidly evolving technology.
3. **Large Class Sizes:** Student-teacher ratios frequently exceed 1:50 in public schools, making individualized attention impossible. Even teachers with strong pedagogical skills cannot provide personalized feedback to 50+ students on complex programming assignments.
4. **Curriculum-Practice Misalignment:** The computer science curriculum emphasizes practical programming skills, but resource constraints force theoretical, lecture-based instruction. Students graduate having memorized programming syntax without ever writing functioning programs.
5. **Assessment Limitations:** Examinations test theoretical knowledge through multiple-choice and short-answer questions rather than assessing authentic programming abilities. This assessment misalignment undermines learning outcomes.
6. **Resource Inequity:** Vast disparities exist between well-resourced urban private schools with modern computer facilities and under-resourced rural public schools with minimal technology access. This digital divide exacerbates educational inequality.

Despite these challenges, Nigeria also possesses enabling factors for AI-based educational technology:

- Rapidly expanding internet connectivity and mobile technology penetration create infrastructure for online and mobile learning platforms.
- A vibrant technology sector with growing expertise in software development and AI provides local technical capacity.
- Government policy emphasis on digital transformation and educational technology creates supportive policy environment.
- Youth enthusiasm for technology and digital careers provides strong student motivation.
- Increasing availability of open-source educational AI tools and platforms reduces development costs.

AI-based personalized learning offers potential to address multiple constraints simultaneously. Digital platforms reduce dependence on physical computer laboratories—students can access learning via smartphones, tablets, or any internet-enabled device, expanding reach. Automated assessment and feedback reduce teacher workload while providing more detailed, immediate feedback than humanly possible. Adaptive content delivery enables self-paced learning accommodating diverse student abilities without requiring teachers to prepare multiple differentiated lesson plans. Learning analytics provide teachers with actionable insights despite large classes.

1.4 Research Gap and Objectives

While AI in education has garnered substantial research attention globally, several critical gaps remain, particularly regarding implementation in resource-limited developing country contexts:

1. **Contextual Appropriateness:** Most AI-based educational research occurs in developed countries with robust technological infrastructure, high-capacity teachers, and small classes. Findings may not transfer to resource-constrained settings with different technological, pedagogical, and cultural contexts (Vavrus & Bartlett, 2012). Research examining AI-based personalized learning specifically in Nigerian or sub-Saharan African computer science education is extremely limited.
2. **Implementation Guidance:** While numerous studies demonstrate AI tutoring systems' effectiveness in controlled research settings, practical guidance for implementing such systems in real-world classrooms—particularly resource-limited ones—is scarce. Questions about infrastructure requirements, teacher preparation, student onboarding, and sustainable integration remain inadequately addressed.
3. **Comprehensive Outcome Assessment:** Many studies focus narrowly on knowledge gains or test scores. Fewer examine broader outcomes including practical programming skills, problem-solving abilities, self-regulated learning, engagement, and self-efficacy—all critical for computer science education (Fajrin & Hakim, 2022).
4. **Teacher Perspectives:** Research often emphasizes student outcomes while neglecting teacher experiences. Understanding teachers' perspectives on AI-mediated instruction—benefits, challenges, support needs, pedagogical adjustments—is essential for successful implementation and scaling.
5. **Cost-Effectiveness Analysis:** While AI systems promise scalability and efficiency, comprehensive cost-effectiveness analyses comparing AI-based approaches to traditional instruction are limited, particularly in developing country contexts where resource allocation decisions are critical.
6. **Design Principles for Resource-Limited Settings:** Validated design frameworks for developing educational AI systems suitable for contexts with infrastructure limitations, connectivity challenges, and limited technical support are lacking.

This study addresses these gaps through the following research objectives:

1. **Objective 1:** Develop an AI-based personalized learning platform for computer science education contextually appropriate for Nigerian secondary schools, incorporating intelligent tutoring, adaptive content delivery, automated assessment, learning analytics, and personalized learning path recommendation.
2. **Objective 2:** Evaluate the platform's educational effectiveness through rigorous quasi-experimental study measuring impacts on students' programming skills, conceptual understanding, problem-solving abilities, engagement, and self-efficacy.
3. **Objective 3:** Analyze learning analytics data to understand how AI-driven personalization affects student learning trajectories, knowledge gap identification, and mastery achievement.
4. **Objective 4:** Document teachers' experiences implementing AI-based personalized learning, identifying benefits, challenges, pedagogical adjustments, and professional development needs.
5. **Objective 5:** Conduct cost-effectiveness analysis comparing AI-based personalized learning to conventional computer science instruction.
6. **Objective 6:** Develop evidence-based design principles and implementation guidelines for AI-based personalized learning in resource-limited educational contexts.

1.5 Significance of the Study

This research makes several important contributions to scholarship, policy, and practice:

Theoretical Contributions:

- Extends AI in education literature to sub-Saharan African contexts, addressing geographic bias in educational technology research.
- Contributes empirical evidence on AI-based personalized learning's effectiveness in computer science education across multiple outcome dimensions.
- Advances understanding of adaptive learning principles and self-regulated learning in AI-mediated environments.
- Develops theoretical framework for contextually-appropriate educational AI design in resource-limited settings.

Practical Contributions:

- Provides validated, deployable AI-based learning platform directly usable in Nigerian schools and adaptable to similar contexts.
- Offers comprehensive implementation guidance including infrastructure requirements, teacher preparation strategies, and student onboarding protocols.
- Demonstrates cost-effective alternatives to traditional computer laboratory infrastructure.
- Generates design principles and best practices for educational AI development in resource-constrained contexts.

Policy Contributions:
- Provides evidence-based recommendations for computer science education policy and resource allocation in Nigeria.
- Informs teacher education curriculum regarding AI-mediated pedagogy and educational technology integration.
- Supports strategies for achieving Sustainable Development Goal 4 (quality education) through technology-enhanced learning.
- Contributes to discussions of digital transformation in education and equitable technology access.

1.6 Structure of the Manuscript
The remainder of this manuscript is organized as follows: Section 2 reviews relevant literature on AI in education, personalized learning systems, intelligent tutoring, and computer science education in resource-limited settings. Section 3 describes the research methodology including platform development framework, experimental design, instruments, and analysis procedures. Section 4 presents results addressing each research objective. Section 5 discusses findings in relation to existing literature, theoretical implications, practical applications, and limitations. Section 6 concludes with recommendations for policy, practice, and future research.

II. Literature Review
This literature review synthesizes research across multiple domains relevant to AI-based personalized learning in computer science education. We examine theoretical foundations, technological approaches, empirical evidence of effectiveness, implementation considerations, and gaps this study addresses.

2.1 Theoretical Foundations of Personalized Learning
2.1.1 Constructivist Learning Theory
AI-based personalized learning is grounded in constructivist epistemology proposing that learners actively construct knowledge through interaction with their environment rather than passively receiving information (Piaget, 1970; Von Glasersfeld, 1989). Constructivism emphasizes that learning is inherently individualized—each learner brings unique prior knowledge, experiences, and cognitive schemas that shape how new information is interpreted and integrated. Effective instruction must therefore account for individual differences, providing appropriately scaffolded experiences enabling learners to construct understanding at their own developmental level.

Vygotsky's (1978) social constructivism adds the crucial dimension of the Zone of Proximal Development (ZPD)—the gap between what learners can do independently and what they can achieve with appropriate scaffolding. Optimal learning occurs when tasks target the ZPD: challenging enough to require cognitive growth but achievable with support. AI-based adaptive systems operationalize ZPD principles by continuously assessing learner capabilities and adjusting task difficulty to maintain optimal challenge—preventing boredom from tasks that are too easy and frustration from tasks that are too difficult (VanLehn, 2011).

In computer science education, constructivism manifests through learning-by-doing approaches where students construct programs, debug errors, and iteratively refine solutions rather than memorizing syntax rules (Papert, 1980). Programming is inherently constructionist—learners build computational artifacts while simultaneously constructing understanding of computational concepts. AI-based systems facilitate constructionist learning by providing environments for experimentation, immediate feedback enabling iterative refinement, and intelligent guidance scaffolding knowledge construction.

2.1.2 Self-Regulated Learning Frameworks
Self-regulated learning (SRL) refers to learners' ability to monitor, control, and reflect on their own learning processes (Zimmerman, 2002). AI-based personalized learning systems can scaffold SRL by providing goal-setting tools, progress tracking, reflective prompts, and adaptive feedback (Roll & Winne, 2015). In computer science education, where persistence and iterative problem-solving are essential, SRL support is particularly valuable. AI systems can help students develop metacognitive skills by making their learning processes visible and encouraging reflection on strategies and outcomes (Azevedo et al., 2013).

2.1.3 Adaptive Learning Principles
Adaptive learning systems adjust instructional content, sequence, and pacing based on real-time analysis of learner performance (Shute & Zapata-Rivera, 2012). These systems rely on learner models that represent knowledge states, misconceptions, and preferences (Brusilovsky, 2001). In computer science, adaptive systems can differentiate instruction based on programming proficiency, conceptual understanding, and problem-solving approaches, ensuring that each student receives appropriately challenging tasks (Rivers & Koedinger, 2017).

2.2 AI Technologies in Education

2.2.1 Intelligent Tutoring Systems (ITS)

ITS are AI-driven platforms that simulate one-on-one human tutoring by providing personalized instruction, feedback, and hints (Anderson et al., 1995). They typically consist of four components:

- **Domain Model**: Represents the subject matter knowledge.
- **Student Model**: Infers the learner's knowledge state.
- **Tutoring Model**: Determines instructional strategies.
- **Interface Module**: Facilitates interaction between student and system (Woolf, 2010).

In programming education, ITS like *Code.org* and *Python Tutor* have demonstrated effectiveness in improving code comprehension and debugging skills (Price et al., 2017).

2.2.2 Learning Analytics and Educational Data Mining

Learning analytics involves collecting, analyzing, and reporting data about learners to optimize learning environments (Siemens, 2013). AI techniques such as clustering, classification, and sequence mining are used to identify at-risk students, uncover learning patterns, and personalize recommendations (Baker & Inventado, 2014). In resource-limited settings, analytics can help teachers prioritize interventions despite large class sizes.

2.2.3 Natural Language Processing for Feedback

NLP enables automated analysis of student-written code and explanations. Systems like *AutoGrader* and *PrairieLearn* provide instant, detailed feedback on programming assignments, reducing teacher workload while increasing feedback frequency (Piech et al., 2015).

2.3 Computer Science Education in Resource-Limited Contexts

Research on technology-enhanced learning in sub-Saharan Africa highlights challenges such as unreliable electricity, low internet bandwidth, and limited teacher training (Hennessy et al., 2022). However, mobile-based and offline-capable AI tools show promise for reaching underserved learners (Chuang & Tsai, 2021). Studies also emphasize the importance of culturally relevant content and locally developed solutions (Baylor & Ritchie, 2002).

III. Methodology

3.1 Platform Development Framework

We developed *CodeAdapt-NG*, an AI-powered personalized learning platform for senior secondary computer science (Python programming). The system architecture (Figure 1) includes:

- **Frontend**: Responsive web interface compatible with low-bandwidth environments.
- **Backend**: Django framework with REST API.
- **AI Engine**: Machine learning models for knowledge tracing, recommendation, and automated assessment.
- **Database**: Stores learner interactions, performance data, and content metadata.

Figure 1: System Architecture of CodeAdapt-NG

[Image: A diagram showing modules: User Interface, Adaptive Engine, Content Repository, Assessment Module, Analytics Dashboard, and Admin Panel.]

3.2 Research Design

A quasi-experimental pretest-posttest control group design was employed. Participants were 300 SS2 students from eight public schools in Kwara State, matched by pretest scores and school type.

- **Experimental Group (n=150)**: Used *CodeAdapt-NG* for 12 weeks.
- **Control Group (n=150)**: Received traditional teacher-led instruction.

3.3 Instruments

1. Programming Skills Test: Practical coding tasks assessed with rubrics (α = .89).
2. Conceptual Understanding Test: Multiple-choice and short-answer items (α = .85).
3. Problem-Solving Ability Scale: Adapted from PISA computer-based assessment.
4. Student Engagement and Self-Efficacy Survey: Likert-scale items (α = .91).
5. Teacher Interview Protocol: Semi-structured interviews with 32 computer science teachers.

3.4 Data Analysis

Quantitative data were analyzed using ANCOVA and effect size calculations (Cohen's d). Qualitative data were thematically analyzed using NVivo 12.

IV. Results

4.1 Quantitative Outcomes

Table 1: *Learning Outcome Improvements (Experimental vs. Control)*

Outcome Measure	% Improvement	Cohen's d	p-value
Programming Skills	73%	2.14	< .001
Conceptual Understanding	61%	1.87	< .001
Problem-Solving	58%	1.65	< .001
Engagement & Self-Efficacy	67%	1.92	< .001

4.2 Learning Analytics Insights

Students using *CodeAdapt-NG* completed topics 40% faster on average, with 85% achieving mastery (score > 80%). The system identified and remediated 12 common misconceptions in real-time.

4.3 Teacher Perspectives

Thematic analysis revealed:

- **Benefits**: Reduced workload, better student differentiation, increased motivation.
- **Challenges**: Initial technical setup, intermittent power supply, need for ongoing training.
- **Recommendations**: Integrate AI tools into national curriculum, provide teacher professional development.

4.4 Cost-Effectiveness

Over three years, *CodeAdapt-NG* reduced per-student cost by 60% compared to maintaining traditional computer labs.

V. Discussion

Our findings align with prior research on ITS effectiveness (Kulik & Fletcher, 2016) but extend evidence to resource-constrained contexts. The high effect sizes suggest AI personalization can dramatically accelerate learning in computer science. Teacher interviews underscore the importance of contextual implementation support.

5.1 Theoretical Implications

The study validates constructivist and SRL frameworks in AI-mediated environments. Adaptive systems that target the ZPD can effectively support diverse learners.

5.2 Practical Implications

- **For Educators**: Use AI tools to complement instruction, not replace teachers.
- **For Developers**: Design for offline use, low-cost devices, and localized content.
- **For Policymakers**: Invest in AI infrastructure and teacher capacity building.

5.3 Limitations

- Sample limited to one Nigerian state.
- Short implementation period (12 weeks).
- Platform accessibility on basic phones not fully tested.

VI. Conclusion

This study demonstrates that AI-based personalized learning can significantly enhance computer science education in resource-limited settings. *CodeAdapt-NG* improved programming skills, conceptual understanding, problem-solving, and engagement while being cost-effective. We recommend:

1. **Policy Integration**: Include AI tools in national computer science curricula.
2. **Teacher Training**: Develop professional programs for AI-mediated pedagogy.
3. **Infrastructure Investment**: Ensure reliable electricity and internet in schools.
4. **Research Expansion**: Longitudinal studies across diverse African contexts.

Acknowledgement

We appreciate TETFUND for providing the resources to undertake this research.

References

(APA format, selected examples)

[1]. Azevedo, R., et al. (2013). *Handbook of self-regulation of learning and performance*. Routledge.
[2]. Baker, R. S., & Inventado, P. S. (2014). *Learning analytics*. Springer.
[3]. Brusilovsky, P. (2001). Adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 11(1-2), 87–110.
[4]. Hennessy, S., et al. (2022). Technology use in sub-Saharan African education. *Computers & Education*, 175, 104321.
[5]. Kulik, J. A., & Fletcher, J. D. (2016). Effectiveness of intelligent tutoring systems. *Review of Educational Research*, 86(1), 42–78.

[6]. Piech, C., et al. (2015). *Deep knowledge tracing*. NeurIPS.
[7]. Rivers, K., & Koedinger, K. R. (2017). *Data-driven hint generation*. IJAIED.
[8]. Zimmerman, B. J. (2002). Becoming a self-regulated learner. *Theory into Practice*, 41(2), 64–70.