Improving Programming Skills of Engineering Students at HBCUs Using AI-enhanced Online Personalized Adaptive Learning Tools

Xishuang Dong, Lijun Qian, Xiangfang Li, Pamela Obiomon
Electrical and Computer Engineering Department
Roy G. Perry College of Engineering
Prairie View A&M University

Yujian Fu, Zhigang Xiao, Shujun Yang, Nelson Barnes
Electrical Engineering and Computer Science Department
Alabama A&M University

Abstract

Future engineering education should be able to train students to master traditional engineering knowledge, as well as to provide high-quality training on advanced techniques, such as data analytics and complex simulations, to meet industrial requirements of interdisciplinary talents. Programming skills are imperative for engineering education and becoming more and more important to training students on innovative techniques in emerging areas such as artificial intelligence (AI) and data science to be competitive workforce for interdisciplinary technology development like Internet of Things (IoT). A team of engineering faculty with complementary expertise from two HBCUs (Prairie View A&M University (PVAMU) and Alabama A&M University (AAMU)) propose to build online AI-enhanced personalized adaptive learning (PAL) tools to enhance engineering education on programming skills at HBCUs. To implement these tools, we plan to complete three tasks with advanced AI techniques: 1) basic online tools that implement sharing learning materials and managing assignments, quizzes, projects, and examinations; 2) PAL path recommender via deep reinforcement learning that recommends PAL paths to learners for maximizing engagement in learning programming, as well as improving corresponding learning performance by selecting items of appropriate difficulty; 3) smart programming assistant (SPA) via deep learning-based language models (LMs) that can generate reference codes to assist learners for programming activities involved in assignments and projects. These tools will be extended for teaching other engineering courses by transfer reinforcement learning, which will not require substantial efforts on system implementation to improve other skills of engineering students.

Introduction

Programming is an imperative component of engineering curricula since it helps prepare future workforce through training in critical thinking, problem-solving, and system designs [1]. Recent years witnessed that online tools played a critical role to enhance STEM education [2]. For instance, the online tools have been employed to enhance education outcomes for various courses such as mathematics, physics and nursing. Furthermore, the COVID-19 pandemic has fundamentally changed the course delivery methods from mainly face-to-face to mixed methods where the online tools become indispensable [3].

Although the programming education has been improved with emerging online tools and advanced pedagogies, it still faces two challenges: 1) lacking engagement during learning programming skills with existing online tools such as Canvas [4] and Blackboard [5], especially when students run into obstacles of learning programming. Specifically, current online tools seem not be able to dynamically and automatically select items of appropriate difficulty for learners through effectively estimating learner's knowledge level. This is critical for HBCU students who may come from financially challenged families without strong programming background. Moreover, the engagement issue became worse during the COVID-19 pandemic for online learning of STEM education [6], and the pandemic has disproportionally affected HBCU students [7]; 2) lacking effective references to students for programming practices, assignments, and projects, especially for freshmen students in engineering education. Programming is an abstract thinking process that involves critical thinking and computational design. Without effective programming references, it is challenging for students to learn programming, as well as for instructors to teach programming.

To overcome these challenges, this paper proposed developing on-line personalized adaptive learning (PAL) tools through integrating state-of-the-art pedagogies with deep learning technologies to enhance programming skills of engineering students, especially at HBCUs. PAL is an emerging pedagogical approach enabled by smart learning environments. AI techniques such as machine learning have been successfully applied to improve the recommendation satisfaction by identifying PAL patterns in online education and automatically modeling relationship between different concepts in the curriculum [8]. The proposed online PAL tools include three components: 1) basic online tools that implement sharing learning materials and managing assignments, quiz, projects, and examinations; 2) PAL path recommender via deep reinforcement learning [9] that recommends PAL paths to learners for maximizing engagement in learning programming, as well as corresponding learning performance by selecting items of appropriate difficulty; 3) smart programming assistant (SPA) via deep learning-based language models (LMs) [10] that can generate reference codes to assist learners in the process of programming activities involved in assignments and projects. These three components will effectively cooperate to improve outcomes for programming education. Additionally, these tools will not be limited to programming education and they would be extended to teaching other engineering courses by transfer reinforcement learning [11].

Related Work

1. Online Tools

Online tools allow students to develop knowledge and skills with the help of lecturers and share learning support tools [12]. The most popular online tools include Blackboard [5], Moodle [13] and Canvas [4] that significantly contributed to education through constant availability and accessibility to course materials, collaboration amongst students and lecturers, improved teaching outcomes, and feedback from users [14], especially during COVID-19 pandemic. Unfortunately, these tools didn't fully consider pedagogical importance in the whole process of teaching-learning, which cannot be adaptive to students' needs, resulting in their negative attitudes toward these tools [15].

2. Personalized Adaptive Learning

Different from many teaching pedagogies such as evidence-based learning, personalized adaptive learning (PAL) keeps track each individual student's progress and tailor the learning path based on each student's current knowledge and needs rather than providing a one-size-fits-all approach [16]. Personalization is highly desired of the modern workforce. Specifically, PAL tools are personalized learning platforms that adapt to student's learning needs, the sequence and difficulty of the task abilities, the time of feedback and students' preferences by monitoring their learning paths via automated feedback cycles [17]. More important is that PAL tools modify the presentation of material through recording small data and using the educational analytics to ensure the individual adaptation. Existing PAL tools such as Course Arc, WileyPLUS, Smart-Sparrow, OLI, Learning Objects, The Open Learning Initiative, and NWEA [118] greatly fostered education development. However, they are not tailored to programming education or engineering students at HBCUs. In addition, emerging AI technologies should be leveraged to enhance PAL.

3. AI Techniques for Personalized Adaptive Learning

AI-enabled learning tools have been developed based on learning analytics and educational data mining techniques, which offers numerous benefits, including improved learning experiences, time flexibility, the provision of timely feedback, flexibility in managing student's learning experiences and fostering student progression [19]. It helped improve teaching performance of various courses such as mathematics, physics, psychology, nursing, natural languages like English and Greek, and programming languages like Java through applying AI techniques such as Bayesian networks [20] and neural networks [21]. Traditional machine learning algorithms are also employed including decision trees [22], evolution algorithms, K-nearest neighbor (KNN), Support Vector Machines (SVMs), and Bayesian Knowledge Tracing (BKT). Moreover, AI-based applications have been developed for online education. Taking computer vision (CV) as an example, image classification based on neural networks was used to detect non-verbal behaviors of learners [23]. Moreover, recent years witnessed the development of AI-based code generators that have great potentials in assisting programming education [24]. It still, however, faces two challenges in the programming education for engineering: 1) existing PAL tools have not fully considered how to increase engagement for engineering; 2) programming assistants have not been developed and embedded in the current PAL tools to enhance learning effectiveness for learners.

Tool Development

This paper proposed to develop comprehensive online tools through combining personalized adaptive learning (PAL) pedagogy and AI techniques, which is able to dynamically tune learning paths during learning programming, as well as to provide coding references for assisting programming activities. The proposed tools are composed of three components: 1) basic online tools that include various tasks that implement the basic functions to meet basic requirements of online teaching and learning; 2) PAL path recommender that employs deep reinforcement learning [25] and recurrent neural networks [26], to recommend PAL paths in terms of knowledge level and knowledge structures; 3) smart programming assistant that is based on deep learning-based LMs [27] to generate reference codes for programming activities such as programming assignments and projects.

1. Basic online tools

This component aims at creating an online learning environment with various basic online tools for fundamental programming subjects. It includes four modules to: 1) develop a set of online learning modules for students to pick up, learn, and master the fundamental programming materials; 2) develop a student assessment module for students to observe the evaluation results of the learning process; 3) develop a course management module for instructors to manage the course materials; 4) develop a system administration module to manager different users. More details were shown in Fig. 1.

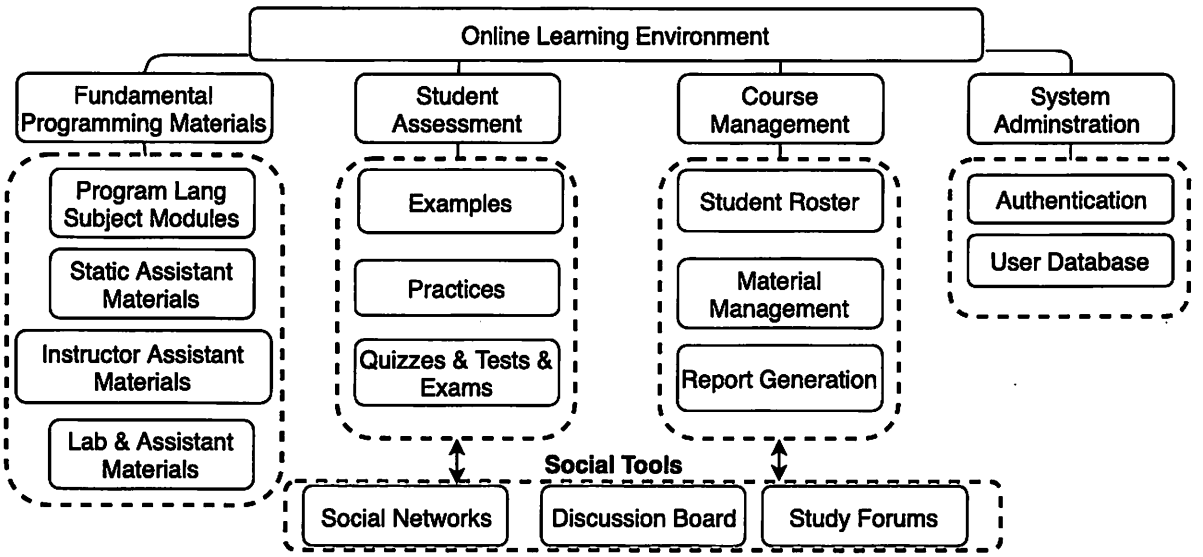

Figure 1. Details of online learning environment.

2. PAL recommender

This component proposed to build PAL path recommender that can sequentially recommend personalized learning items (e.g., lectures, exercises) to satisfy the unique needs of each student. Given historical learning items (or records) of learners H, a learning target T, and a knowledge structure graph G, PAL path recommendation is to recommend a PAL path P with N learning items to maximize the learning effectiveness E of the entire learning path.

Fig. 2 demonstrated the general process of PAL path recommendation. It is supposed that one engineering course requires programming skills for assignments and/or projects, where the graph of knowledge structure demonstrated the prerequisite relations of learning items in the learning process for this course. For different learners, it is to recommend different PAL paths according to different historical learning records of different learners. For example, the PAL path for learner *1* differs from that of learner *n* since the learner *1* has

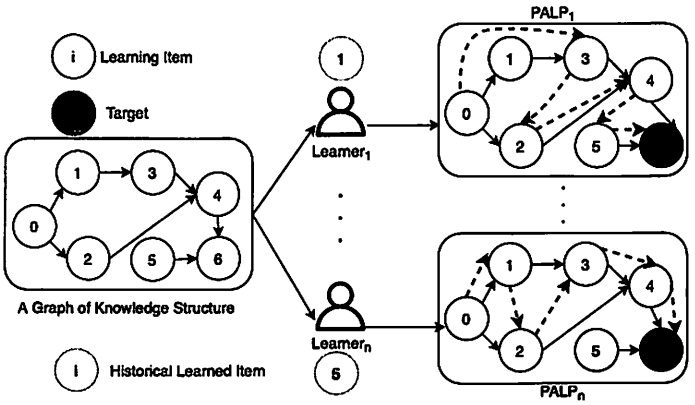

Figure 2. Flow of recommending PAL paths for different learners (students). PALP denotes PAL path that is illustrated with red dot-lines.

completed learning item 1 while the learner *n* has completed learning item 5 prior to joining this course. Moreover, it aims at tuning the learning path based on learning effectiveness step by step to maximum the learning effectiveness of the whole learning process. To implement this component, we view PAL path recommendation as a reinforcement learning [28] task that is to learn a policy to recommend personalized learning path based on the knowledge level and the knowledge structure of programming education, where the knowledge level reflects the masteries on learning items.

3. Smart programming assistant

Learning programming for engineering students is a challenging task since it requires the students to be familiar with tools of coding and debugging, as well as be able to convert the task into mathematical processes of solutions to special problems. Specifically, due to lacking training on mathematical abstracted thinking, the students usually suffer from learning programming without assistants, especially when they are required to complete programming assignments independently. For example, when a student was assigned a project to build a calculator with C/C++ programming language, it was challenging to convert the entire process to comprehensive mathematical processes, which prevented the student from completing programs effectively, even delayed the learning progress. In addition, another challenge is that no effective free tools can help them complete missing key items in the process of programming.

To overcome these two challenges, this component aims to build a smart programming assistant (SPA) enhanced with AI techniques, which can complete missing items of programs and generate reference codes in terms of task description. Fig. 3 presented the workflow of the proposed SPA. When a learner (an engineering student) has issues of incomplete codes, the learner can send the request to the SPA to complete the codes. Moreover, if the learner has no clue for programming tasks, the learner can send the tasks described with natural language to the SPA to synthesize the reference codes, where the tasks should not be complicated as a complex real-world application since the SPA assists programming education, not builds real applications.

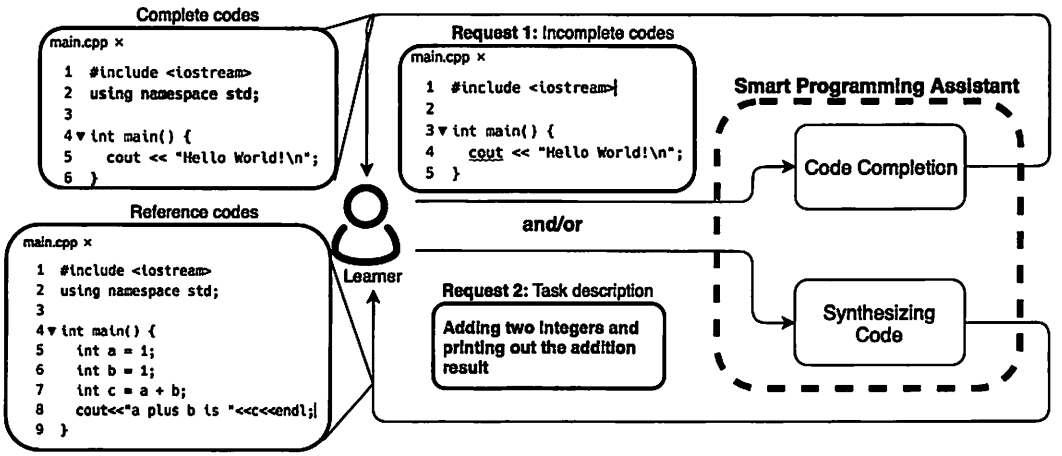

Figure 3. Workflow of smart programming assistant (SPA).

Deep learning-based language models (LMs) [29] have shown impressive performance in modeling source codes, written in programming languages [30], which contributes to downstream tasks like

code completion [52] and synthesizing codes from natural language descriptions [31]. LMs-based code modeling can be classified into three categories: 1) left-to-right LMs; 2) masked LMs; 3) encoder-decoder LMs. Left-to-right LMs, such as CodeGPT [32], Codex [33] and GPT-NeoX [34], predicted the probability of a missing token in terms of previous tokens, which is useful for code completion. Masked LMs, such as CodeBERT [10] and CuBERT [35], predicted masked text pieces based on surrounding context. Encoder-decoder LMs such as CodeT5 [36] and PLBART [37] were developed for code generation with respect to natural language, where an encoder is built to represent an input sequence shown with natural language, and a decode is to generate the codes conditioned on the input sequence.

We planned to study and implement two modules below to develop the SAP enhanced with deep learning-based language models (LMs) [29] techniques:

- Code completion: it is to build a tool to complete codes during the process of programming for engineering students. Specially, it is built based on PolyCoder that is a large neural language model with 2.7B parameters pre-trained with GPT-2 architecture [39] on various programs like C, C++ and Python.

- Code generation: it aims at generating reference codes for simple tasks described with natural language. We applied CodeT5 [36], a unified pre-trained encoder-decoder transformer model, to implement code generation, which seamlessly supports both code understanding and generation tasks. Furthermore, human-in-the-loop techniques [38] will be introduced to enhance code generation quality.

Case Study

Programming skills of engineering students for "ELEG 1301 Programming for Computer Engineering I" at PVAMU will be enhanced with the proposed tools. "ELEG 1301" introduces fundamentals of C/C++ programming language and basic skills of Matlab, including logic of algorithms, flowcharts, program looping, conditional statements, arrays, inputs, outputs, functions and pointers, and Matlab skills. In addition, it also trains students to build engineering applications through team projects. This course aims at providing students with a carefully paced introduction to the basics of C/C++ programming. Students will develop programming skills to resolve engineering problems through logically thinking to arrive at clearly coded algorithms for the final solutions. Two key outcomes of this course include: 1) developing C/C++ programming skills to solve engineering problems through logically thinking; 2) engaging students in team projects which provide a forum to interact and share ideas or articulate positively the merit of alternative solutions in an earnest effort to complete assigned task on time.

The proposed tools will enhance these outcomes effectively. The first outcome will be improved with PAL path recommender and the smart programming assistant (SPA) since PAL path recommender is able to provide PAL paths for "ELEG 1301" in terms of students' knowledge background and the prerequisite relations between learning items in this course. Moreover, SPA can provide assistance on coding by completing codes and synthesizing reference codes when students have challenges on programming for assignments. For the second outcome, the basic online tools can help share learning resources such as slides and course videos, and enhance the student's engagement through convenient

and effective communications via embedded social tools. These three components of the proposed tools can effectively enhance learner's (student's) engagement for this course.

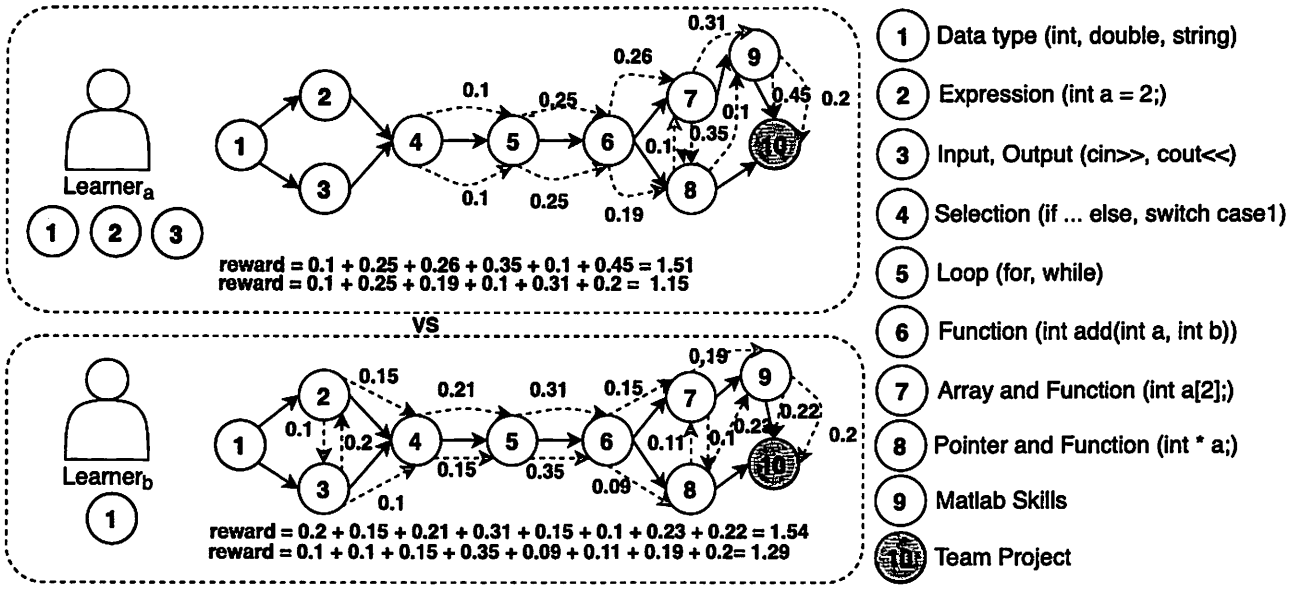

Figure 4. An example of recommending PAL paths for two different learners (students) involved in ELEG 1301 including 9 learning sections for learning C/C++ programming. The green circles denote learning sections completed prior to joining ELEG 1301 and the red circles denotes the learning target. The red and blue dot-lines denotes two possible learning paths with different rewards that are built based on the performance for completing learning sections on these paths.

Fig. 4 showed an example of recommending PAL paths for different learners involved in "ELEG 1301" with the identical knowledge structure. Learner *a* completed section 1, 2, and 3 while learner *b* only completed section 1 before they joined "ELEG 1301". The PAL path recommender is able to build different learning paths regarding their historical learning records through maximizing rewards via reinforcement learning. For example, it presented two possible PAL paths marked with red and blue dot-lines for learner *a*, and the PAL path recommender will suggest the red path with larger rewards for this learner. Moreover, Fig. 5 demonstrated the process of applying SPA to help a student team for complete the team project of "ELEG 1301". The student team could present the questions on the tasks of the team project with natural language to SPA. Through the code generation module, SPA will be able to produce reference codes to these questions, which helps the student team accomplish the team project.

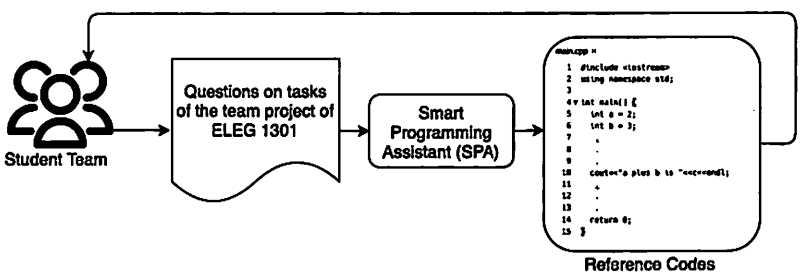

Figure 5. Enhancing team project of ELEG 1301 via SPA

Conclusion

This paper proposed to build a set of online tools to enhance programming skills for engineering students for STEM education through integrating AI techniques and personalized adaptive learning pedagogies. These tools enable engineering students to share learning materials, recommend learning

contents regarding students' background and learning curves, and generate coding references during completing programming assignments. We presented a case study with programming course ELEG 1301 at PVAMU to demonstrate how to apply these tools to programming education. Moreover, these tools have great potentials to be extended to enhance education for other courses. In the future, we will present detailed implementation and validation of these tools through education activities of programming for various programming languages such as C/C++ and Python.

References

1. Topalli, D., Cagiltay, N. E. 2018. Improving programming skills in engineering education through problem-based game projects with Scratch. Computers & Education, 120, 64-74.
2. Arici, F., Yildirim, P., Caliklar, Ş., Yilmaz, R. M. 2019. Research trends in the use of augmented reality in science education: Content and bibliometric mapping analysis. Computers & Education, 142, 103647.
3. Giatman, M., Siswati, S., & Basri, I. Y. 2020. Online learning quality control in the pandemic Covid-19 era in Indonesia. Journal of Nonformal Education, 6(2), 168-175.
4. https://www.instructure.com/canvas.
5. https://www.blackboard.com.
6. Hollister, B., Nair, P., Hill-Lindsay, S., Chukoskie, L. 2022. Engagement in Online Learning: Student Attitudes and Behavior During COVID-19. In Frontiers in Education (Vol. 7, p. 851019). Frontiers Media SA.
7. Mostafa, S., Cousins-Cooper, K., Tankersley, B., Burns, S., Tang, G. 2022. The impact of COVID-19 induced emergency remote instruction on students' academic performance at an HBCU. Plos one, 17(3), e0264947.
8. Chaplot, D. S., Rhim, E., Kim, J. 2016. Personalized adaptive learning using neural networks. In Proceedings of the third (2016) ACM conference on learning@ scale (pp. 165-168).
9. Arulkumaran, K., Deisenroth, M. P., Brundage, M., Bharath, A. A. 2017. Deep reinforcement learning: A brief survey. IEEE Signal Processing Magazine, 34(6), 26-38.
10. Feng, Z., et al. 2020. Codebert: A pre-trained model for programming and natural languages. arXiv preprint arXiv:2002.08155.
11. Taylor, M. E., Stone, P. 2009. Transfer learning for reinforcement learning domains: A survey. Journal of Machine Learning Research, 10(7).
12. Gros, B. (2016). The design of smart educational environments. Smart learning environments, 3(1), 1-11.
13. https://moodle.org.
14. Pappas, I. O., Giannakos, M. N. 2021. Rethinking learning design in IT education during a pandemic. In Frontiers in education (Vol. 6, p. 652856). Frontiers Media SA.
15. Kabudi, T., Pappas, I., Olsen, D. H. 2021. AI-enabled adaptive learning systems: A systematic mapping of the literature. Computers and Education: Artificial Intelligence, 2, 100017.
16. Xie, H., Chu, H. C., Hwang, G. J., Wang, C. C. 2019. Trends and development in technology-enhanced adaptive/personalized learning: A systematic review of journal publications from 2007 to 2017. Computers & Education, 140, 103599.
17. Pliakos, K., Joo, S. H., Park, J. Y., Cornillie, F., Vens, C., Van den Noortgate, W. 2019. Integrating machine learning into item response theory for addressing the cold start problem in adaptive learning systems. Computers & Education, 137, 91-103.
18. Osadcha, K., Osadchyi, V., Semerikov, S., Chemerys, H., Chorna, A. 2020. The review of the adaptive learning systems for the formation of individual educational trajectory. CEUR Workshop Proceedings.
19. Moreno-Guerrero, A. J., López-Belmonte, J., Marín-Marín, J. A., Soler-Costa, R. 2020. Scientific development of educational artificial intelligence in Web of Science. Future Internet, 12(8), 124.
20. Jensen, F. V. 1996. An introduction to Bayesian networks (Vol. 210, pp. 1-178). London: UCL press.
21. LeCun, Y., Bengio, Y., Hinton, G. 2015. Deep learning. nature, 521(7553), 436-444.
22. Safavian, S. R., Landgrebe, D. 1991. A survey of decision tree classifier methodology. IEEE transactions on systems, man, and cybernetics, 21(3), 660-674.
23. Holmes, M., Latham, A., Crockett, K., O'Shea, J. D. 2017. Near real-time comprehension classification with artificial neural networks: Decoding e-learner non-verbal behavior. IEEE Transactions on Learning Technologies, 11(1), 5-12.

24. Ling, W., Grefenstette, E., Hermann, K. M., Kočiský, T., Senior, A., Wang, F., Blunsom, P. 2016. Latent predictor networks for code generation. arXiv preprint arXiv:1603.06744.
25. Liu, Q., Tong, S., Liu, C., Zhao, H., Chen, E., Ma, H., Wang, S. 2019. Exploiting cognitive structure for adaptive learning. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (pp. 627-635).
26. Salehinejad, H., Sankar, S., Barfett, J., Colak, E., Valaee, S. 2017. Recent advances in recurrent neural networks. arXiv preprint arXiv:1801.01078.
27. Raychev, V., Vechev, M., Yahav, E. 2014. Code completion with statistical language models. In Proceedings of the 35th ACM SIGPLAN Conference on Programming Language Design and Implementation (pp. 419-428)
28. Sutton, R. S., Barto, A. G. 2018. Reinforcement learning: An introduction. MIT press.
29. Brown, T., et al. 2020. Language models are few-shot learners. Advances in neural information processing systems, 33, 1877-1901.
30. Alon, U., Sadaka, R., Levy, O., Yahav, E. 2020. Structural language models of code. In International conference on machine learning (pp. 245-256). PMLR.
31. Desai, A., Gulwani, S., Hingorani, V., Jain, N., Karkare, A., Marron, M., Roy, S. 2016. Program synthesis using natural language. In Proceedings of the 38th International Conference on Software Engineering (pp. 345-356).
32. Lu, S., et al. 2021. Codexglue: A machine learning benchmark dataset for code understanding and generation. arXiv preprint arXiv:2102.04664.
33. Chen, M., et al. 2021. Evaluating large language models trained on code. arXiv preprint arXiv:2107.03374.
34. Black, S., et al. 2022. Gpt-neox-20b: An open-source autoregressive language model. arXiv preprint arXiv:2204.06745.
35. Kanade, A., Maniatis, P., Balakrishnan, G., Shi, K. 2020. Learning and evaluating contextual embedding of source code. In International Conference on Machine Learning (pp. 5110-5121). PMLR.
36. Wang, Y., Wang, W., Joty, S., Hoi, S. C. 2021. Codet5: Identifier-aware unified pre-trained encoder-decoder models for code understanding and generation. arXiv preprint arXiv:2109.00859.
37. Ahmad, W. U., Chakraborty, S., Ray, B., Chang, K. W. 2021. Unified pre-training for program understanding and generation. arXiv preprint arXiv:2103.06333.
38. Zanzotto, F. M. 2019. Human-in-the-loop artificial intelligence. Journal of Artificial Intelligence Research, 64, 243-252.
39. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I. 2019. Language models are unsupervised multitask learners. OpenAI blog, 1(8), 9.

XISHUANG DONG

Xishuang Dong is Assistant Professor of Electrical and Computer Engineering Department, Roy G. Perry College of Engineering, Prairie View A&M University. His research interests include deep learning, object detection, natural language processing, computer systems biology, and Internet of Things.

LIJUN QIAN

Lijun Qian is Professor of Electrical and Computer Engineering Department, Roy G. Perry College of Engineering, Prairie View A&M University. His research interests are in the area of big data processing, artificial intelligence, wireless communications and mobile networks, network security and intrusion detection, and computational systems biology.

XIANGFANG LI

Xiangfang Li is Associate Professor of Electrical and Computer Engineering Department, Roy G. Perry College of Engineering, Prairie View A&M University. Her research interests are in computational and systems biology, systems pharmacology, computer networking and communication, and artificial intelligence. She is the recipient of the Outstanding Researcher of the Year award from the Roy G. Perry College of Engineering of PVAMU in 2017.

PAMALA OBIOMON

Pamela Obiomon is Dean of Roy G. Perry College of Engineering, Prairie View A&M University. Her research interests are in the areas of integrated microsystems for environmental sensing powered by energy scavenging, smart systems using field-programmable gate arrays (FPGAs), and the design of FPGA-based controllers for autonomous vehicles.

YUJIAN FU
Yujian Fu is Professor of Electrical Engineering and Computer Science Department, Alabama A&M University. Her research interests are in the areas of formal specification, verification and validation, optimization, reinforcement learning, mobile security, behavior based malware detection, specification based intrusion detection, formal specification & verification of cyber physical systems, real time embedded software design and quality assurance, software testing, runtime verification, software architecture, service oriented architecture (SOA), software evolution.

ZHIGANG XIAO
Zhigang Xiao is Professor of Electrical Engineering and Computer Science Department, Alabama A&M University. His research interests are in the areas of microelectronics/VLSI circuits and microfabrication and nanofabrication.

SHUJUN YANG
Shujun Yang is Assistant Professor of Electrical Engineering and Computer Science Department, Alabama A&M University. His research interests are in the areas of Bandstop Filter and Simulation.

NELSON BARNES
Nelson Barnes is Assistant Professor of Electrical Engineering and Computer Science Department, Alabama A&M University. His research interests are in the areas of Mobile Device Development, Neural Networks and Machine Learning, Robotics and Autonomous Vehicles (Ground, Marine, Aerial), Software Engineering, Software Project Management, Information Systems Analysis, and Information Systems Security.