A Random Forest Model for Personalized Learning in a Narrative Game

Dr. Ying Tang, Rowan University

Ying Tang received the B.S. and M.S. degrees from the Northeastern University, P. R. China, in 1996 and 1998, respectively, and Ph.D degree from New Jersey Institute of Technology, Newark, NJ, in 2001. She is currently a Professor of Electrical and Computer Engineering (ECE) at Rowan University, Glassboro, NJ. Her research interests include virtual reality and augmented reality, artificial intelligence, and modeling and scheduling of computer-integrated systems. Dr. Tang is very active in adapting and developing pedagogical methods and materials to enhance engineering education. Her most recent educational research includes the collaboration with Tennessee State University and local high schools to infuse cyber-infrastructure learning experience into the pre-engineering and technology-based classrooms, the collaboration with community colleges to develop interactive games in empowering students with engineering literacy and problem-solving, the integration of system-on-chip concepts across two year Engineering Science and four year ECE curricula, and the implementation of an educational innovation that demonstrates science and engineering principles using an aquarium. Her work has resulted in over 100 journal and conference papers and book chapters.

Mr. Ryan Hare, Rowan University

Ryan Hare received his B.S. in Electrical and Computer Engineering from Rowan University in 2019. He is currently pursuing his M.S. in Electrical and Computer Engineering at Rowan University. His current research focus is applying machine learning and games to enhance student education, particularly in STEM fields.

Random Forest-Based Adaptive Narrative Game for Personalized Learning

Abstract

In student education, learning styles can vary wildly from one student to the next. While students should receive support tailored to their specific learning style, this type of personalized support can often not be realized due to resource constraints. This paper presents an implementation of personalized learning support utilizing a random forest machine learning model built on top of an existing narrative game environment. The existing game, Gridlock, is a domain-specific narrative game that implements metacognitive strategies to assist students in learning sequential logic design, a core topic in Computer Engineering and Computer Science. The metacognitive strategies featured in the game are Roadmap, What I Know-What I Want to Know-What I Need to Solve (KWS), and Think-Aloud-Share-Solve (TA2S). Roadmap provides students with an idea of what they have learned and what they still need to learn. KWS prompts students to remember what they already know, what they want to know, and what they are trying to solve to keep them focused and on task. TA2S encourages students to think aloud by communicating with fellow classmates to share their solutions and collaborate to solve problems.

On top of existing learning strategies within the game, a random forest machine learning model is used to classify students into various categories based on their learning style. To train this model, a large dataset was generated based on previously gathered information from tests of the game as well as in-classroom observations of students playing through the game. The model was verified through multiple runs with students of varying levels of subject knowledge. As they play through the game, students are classified based on their perceived knowledge of the subject matter presented to them. From this classification, students can be provided individualized assistance in the form of tutorials, hints, prompts, or even videos of experts solving similar problems. These tailored prompts provide students with immediate feedback in their areas of difficulty, maintaining the momentum of the learning process and improving student comprehension.

Introduction

With recent efforts in student education placing major focus on student knowledge transference and problem solving[1], problem-based learning (PBL) has gained momentum[2]. This is especially true for more complicated educational paths such as STEM fields; particularly engineering. PBL focuses on pushing students to develop and apply their own problem-solving processes when given a new problem. While PBL methods have been successfully implemented in engineering education[3,4,5], these methods can also hamper a student's learning if the student prefers a more structured educational process. The negative aspect also applies to students who lack the necessary prior knowledge or motivation to explore and learn without prompting. As such, a ”one-size-fits-all” approach to PBL is not the correct path when teaching large numbers of diverse engineering students[6].

The long-standing schism between direct student guidance and student discovery-focused learning has been a major topic of research[7,3]. However, there is little research present that attempts to deviate from this ”one-size-fits-all” system[8,9]. The National Academy of Engineering has even listed personalized student education as one of the 14 Grand Challenges for the 21st century[10].

As such, the logical step forward from current research is to create a system that can provide tailored assistance based on an interpretation of a student’s individual learning methods and knowledge. Attempts at these types of systems have been made in the form of Intelligent Tutoring Systems (ITS)[11], and these systems do provide a level of guidance to students. However, they often lack engagement and are not well implemented in PBL situations. Further, effective systems often still require instructor interference or in some way rely upon instructor interaction.

To deliver such a system, a narrative game offers one potential vector. These types of games can be used to provide players with goal-focused features while simulating real-world problems. Much research has demonstrated that these educational narrative games can both support student cognitive development[12,13,14,15,16,17] and provide benefits in student assessment[18,19,20].

The proposed system builds on top of a previously created narrative game[21,16] and is referred to as a Personalized Instruction and Need-aware Gaming (PING) system. The system integrates an ITS with a PBL-focused educational process within a narrative game. The end goal of the system is to detect learning difficulties from students as they play and provide necessary personalized support to students. While they receive this support, the students are immersed in a simulation of a real-world problem-solving situation. The end result is that any present learning difficulties are addressed immediately, allowing students to focus on improving their domain knowledge, subject comprehension, and problem-solving capabilities. The system also requires little to no instructor interaction, leading to significantly reduced resource requirements for implementation.

The PING system utilizes multi-component probing methods informed by Social Cognitive Learning Theory[22]. A random forest classifier allows the system to determine a student’s required level of support from the output of the probing methods, which constantly collect data s the student plays the game. At several points within the game, the system adjusts the content to fit the student’s areas of difficulty. The game also offers support or prompts to encourage progress within the game. While the overarching problem is the same for every student, the path they take to reach the solution will vary drastically.

The proposed PING system combines techniques of statistical inference, cognitive psychology, education research, sensor informatics, and machine learning techniques to provide students a personalized education process. The contextual problem-solving situation engages students, giving them incentives to succeed in their learning process while allowing them to both be entertained and move at their own pace. Overall, the system will improve student knowledge of engineering concepts while increasing their ability to solve complex problems and extrapolate new knowledge from general principles.

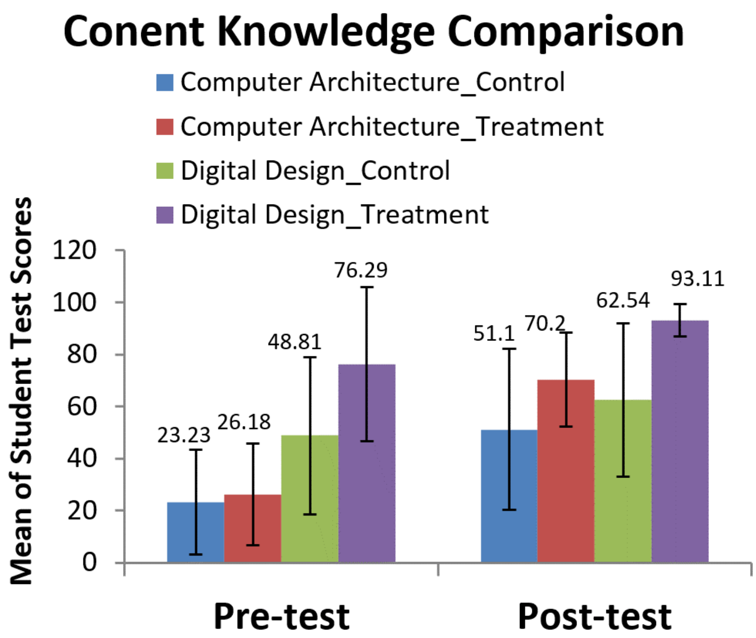

Figure 1: Comparison of content knowledge. Group difference significant (t test $p<0.05$) [Tang *et al. , 2017].*

Modified Existing Game System

Rather than create a new narrative game for the PING system, the system is instead built on top of a pre-existing game called Gridlock. Gridlock is designed to assist students with basic concepts within the subject of digital logic design. Students who play the game find themselves in a position of redesigning a traffic light controller. The traffic light controller is a basic problem within digital logic design as it can easily be represented by a sequential state machine. In the early portion of the game, students are motivated to solve the problem by witnessing the car accident that occurs when the traffic light fails. As the student plays through the sections of the game, they are tested and made to practice several subjects that relate to the final design process.

Gridlock provides a solid starting point for the PING system because it has previously demonstrated success as an educational tool. Previous evaluations of the game had over 300 students in seven courses at Tennessee State and Rowan Universities [17]. Students were given evaluation tests before and after playing the game, and the results are shown in Figure 1 [17]. Treatment sections showed significant improvement over control sections. Furthermore, students described the game as more interesting and engaging over a conventional problem-solving approach to learning [17].

The game already utilizes several meta-cognitive strategies that are meant to improve student learning: 1. Know-Want to know-Solve (KWS): Students are prompted to remember what they know or have already learned. They are also prompted to consider what they still need to learn in order to solve the final problem. 2. Think-Aloud-Share-Solve (TA2S): Students are able to interact through a local chat server, promoting collaboration and intellectual synergy, enhancing

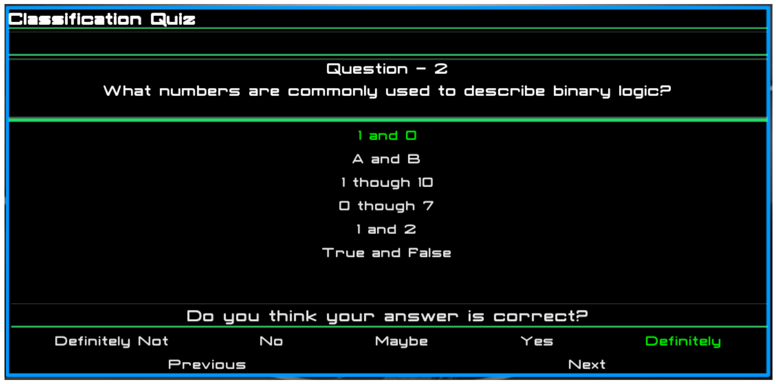

Figure 2: Example of one of the questions used to classify students' initial subject knowledge.

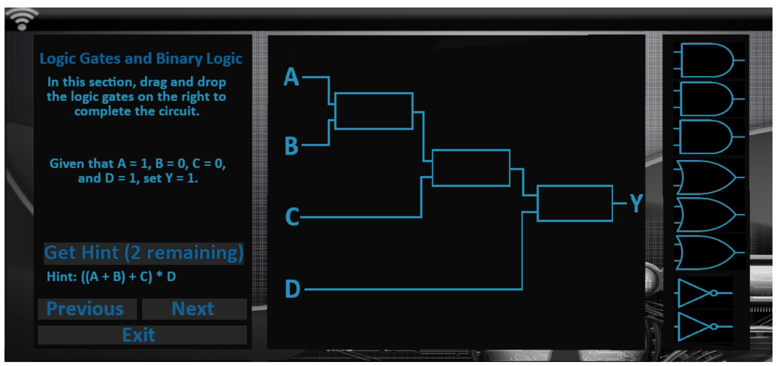

Figure 3: One of the concept-specific sections of the game. Students solve this section by completing the circuit with the correct logic gate.

the learning process. 3. Roadmap: Students are provided with access to in-game study materials that are meant to aid students in locating relevant information[23].

The game begins with a short quiz that provides data to the classification module. The classification module uses the data to provide an initial estimation of the student's level of domain knowledge. This classification is then fed to the path decision module which modifies the later sections of the game in response to the student's initial classification. The modifications are made such that the student is given sections that most closely relate to their areas of difficulty. The later sections of the game are composed of study materials, small problem-solving games, or additional quiz questions to further develop the student's knowledge.

Figure 3 shows one of the specific sections of the game. In this section, students complete a given circuit with logic gates. This helps the student practice their knowledge of logic gates and binary logic.

Upon completing the concept-specific segments of the game, all students then proceed to the final problem-solving stage. The game provides students with a state machine design tool to assist them in their thinking process as they finalize their design. They are then instructed to complete their design in Verilog, a hardware description language. Once complete, they can upload their code into the game, where they are then provided feedback about both syntax errors and logic

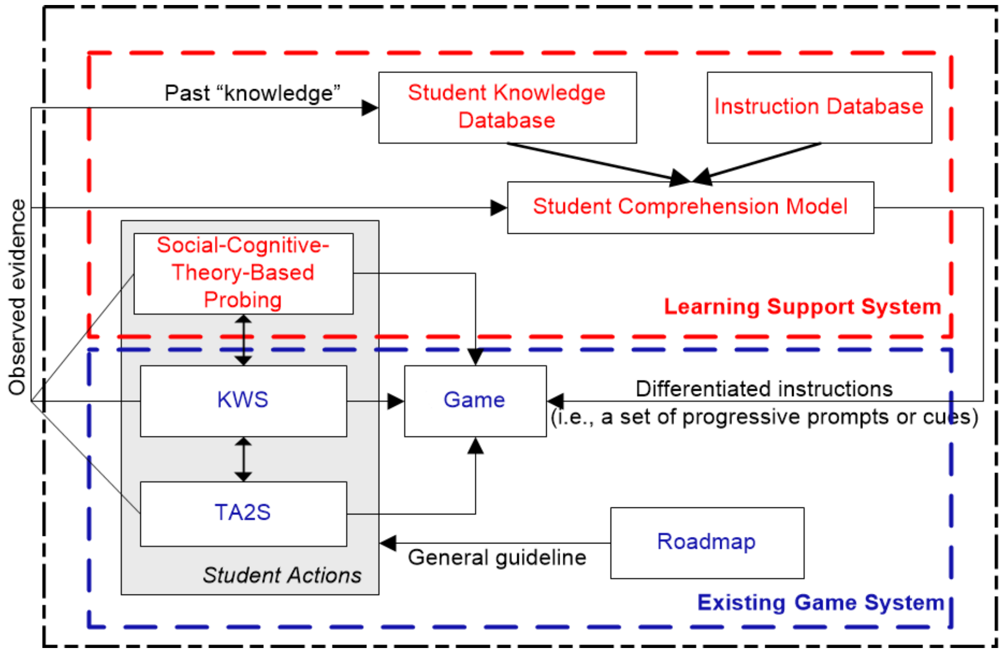

Figure 4: Architecture of the PING system. Components enclosed in blue represent components already within the existing game system. Components in red represent the PING system components to be implemented in the game.

errors within their code.

As students play, their actions are recorded and logged to a remote server. This data can then be used to analyze their performance and improve the game. For example, if a specific section of the game takes a very long time on average, that section would require additional guidance to help students succeed. This data collection and continual development and evolution of the game system is a crucial element to the PING system as a whole.

Personalized Instruction and Need-aware Gaming (PING) System Overview

The PING system is designed to avoid any instructor intervention, operating as a fully automated system. In this way, the system requires reduced resources to implement in an educational setting. This gives the automated system an edge over human tutors. And while human tutors might offer better support than the system, the system should prove similarly effective.

The system provides personalized instruction to students by first predicting their current grasp of the relevant concepts. As each student is unique, a large amount of data is needed on each student for the system to form an accurate model. It is therefor necessary for the system to both record and process all the relevant data as the student plays the game.

Figure 4 shows a diagram of the entire PING system. The four main components that comprise the system are the Student Comprehension Model, the Student Knowledge Database, the Social-Cognitive-Theory-based probing, and the Instruction Database.

The Student Comprehension Model is the component that handles decision-making and student classification. It receives student data from the Student Knowledge Database and uses a random forest classifier to classify the student's level of domain knowledge. From there, it modifies the game to fit that student's areas of difficulty. Finally, in later sections of the game, the Student Comprehension Model uses new data from the concept-specific portions of the game to update its classifications and verify that the student is grasping the content.

The Student Knowledge Database is the center for data collection. It stores all the data gathered from the student, making it available to the Student Comprehension Model as needed. The data from the Social-Cognitive-Theory-based probing including key presses, mouse movements, correct and incorrect answers to prompts, and student emotion data.

The Social-Cognitive-Theory-based probing is responsible for prompting the student with questions and hints, as well as capturing results from game sections. The probing system focuses on four main data points: probes, time, error, and emotion. These are discussed in more detail below. To capture emotion data, the probing system uses a trained facial emotion recognition system to extract student emotions from images captured via webcam. All the data it gathers is stored in the Student Knowledge Database.

The Instruction Database is composed of a large selection of relevant prompts and hints that the probing system can access and use to gather student data. The Database also contains a selection of study materials that the student can access to improve their own knowledge.

Student Comprehension Model

To classify student knowledge, a selection of subjects were created for the problem. There are 7 fairly broad topics which relate to the traffic light design: Digital system basics, logic gates, binary logic, flip-flop circuits, finite state machines, traffic light design specifics, and verilog syntax. For other problems, the number of sections might increase or decrease. The design of the system is flexible and the number of sections can easily be increased or decreased.

For each of the topics related to the problem, the student is asked a few questions to establish the initial estimate of their knowledge. The data gathered from each question includes score, time taken to answer, average emotion data, random or unrelated key presses, sporadic mouse movements, and a self-graded confidence score in each answer. These features are then fed into the random forest classifier that estimates student knowledge on a scale of 1 to 3. A classification of 1 signifies solid subject knowledge, while a classification of 3 signifies that major help is needed.

The random forest classifier is used to discern subtler patterns within the human-centric data gathered by this system. The issue with this approach is that a pool of data is required for testing and training of the system. This data can be difficult to obtain and must first be scored by a human expert. So, the system is first trained with a large pool of generated data. As the system is tested and utilized in classroom settings, the gathered data can be used to update the classifier.

As real data is currently unavailable, a pool of data was generated to train and test the model. The data was generated based on observations from focused testing of the game, as well as assumptions about student learning. To simulate real data, a series of normal distributions were used to generate the synthetic data with these features: score, time, and confidence were all

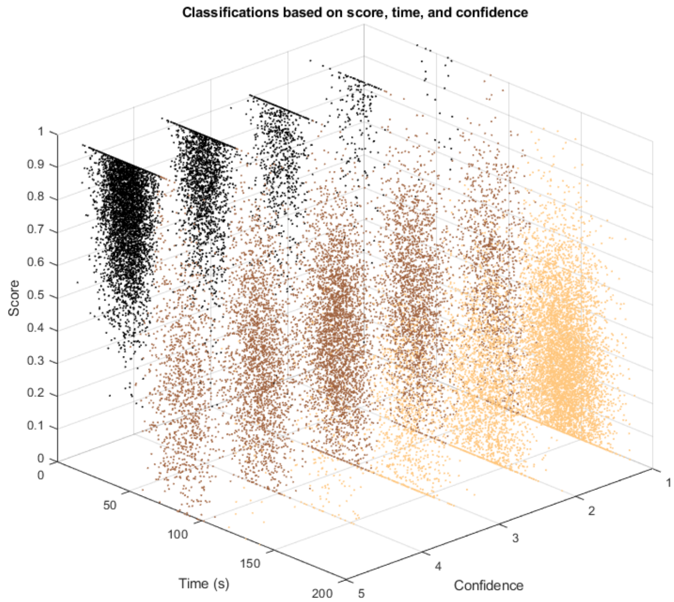

Figure 5: 3D Scatter plot of the classifications based on score, time, and confidence from a quiz. The darkest color is a classification of 1, meaning the student requires little to no help. The lightest color is a classification of 3, meaning the student requires a large amount of help.

generated using a multivariate normal distribution to account for correlations between these variables. Emotion, key presses, and mouse movements were all generated using seperate distributions.

Figure 5 shows a sample of the generated data that was classified by the current iteration of the model. While the plot only shows score, time, and confidence, those three features were found to be the most important when establishing classification.

Many classifiers were well fit for this problem. The basis of a random forest classifier is a number of decision trees. This is desirable for the PING system as decision trees make intuitive decisions. The reasoning behind their classifications can be displayed and easily traced back. Other classifiers, meanwhile, operate more as ”black box” classifiers, making decisions that are not necessarily intuitive. As opposed to a single decision tree, a random forest classifier provides more stability and resistance to data variations and errors[24,25]. These types of ensemble classifiers have also shown improved general performance over single classifiers[26,27].

The random forest classifier also introduces a relatively low computational cost in training. Thus, as new data is gathered, the model can be easily iterated upon to reflect the new data. If new features are found during testing, the model can also be quickly updated to reflect those new features. The random forest classifier then provides a flexible model that can be rapidly retrained to reflect new observations or new data.

Social-Cognitive-Theory-Based Support Integration

An independent educational game environment has disadvantages compared to instructor interaction[28]. Students on the very low end of the content knowledge spectrum might find it impossible to make any progress at all within the game. To remedy this, the student learning supports within the game attempt to replicate traditional instructional practices within the game environment[22]. The goal of the system is to give the students multiple opportunities to learn while simultaneously giving feedback on the errors they've made[29].

The four main components that the supports utilize are probes, time, error, and emotion.

Probes

The PING system utilizes probes by repeatedly asking the student questions to gather more information. The questions pertain to the concept that the student is currently working on, and the resulting data can offer a solid indication of the student's current grasp of the material. These prompts are designed to engage students in a similar fashion to an instructor offering a question to their class. Additionally, by asking the student if they think their answer is correct, their level of confidence can also be measured. This allows for inferences about the student's ability level to be made. For example, a student who gave high confidence on an incorrect answer likely requires some additional prior knowledge. A student who gave low confidence on a correct answer might lack confidence, or might have just guessed the answer. These pop-up questions have been proven effective in encouraging students to self-regulate their learning process, forcing them to focus on important areas of the problem[1].

Time

Time also provides a solid indication of student performance. A student with rough concept knowledge will take longer to think of a solution to a problem. To expand upon this, the system offers the student prompts if they take a significant amount of time to solve a section of the game. These prompts offer hints or information to help the student overcome roadblocks. By combining time with probes, a student who takes a long time on a problem and then answers a probe incorrectly could even be sent back to an earlier section to review more basic concepts.

Errors

The PING system diagnoses student errors on question prompts and within their final traffic light design. When students submit their Verilog code, the system detects any syntax or logic errors within the code and draws the student's attention to these errors. The student is then allowed to fix their errors and resubmit their solution. The feedback-resubmission cycle allows students an iterative process to practice error diagnoses and learn through repetition.

Emotions

With personal computers often having integrated webcams, capturing pictures for additional data is viable. To that end, the PING system uses a facial emotion recognition system to determine student emotions as they play through the game. The emotion recognition system uses a

Number	Pre-Test Grade	Human Score	Class
1	100	1	1
2	98	1	1
3	98	1	1
4	95	2	3
5	90	2	2
6	85	2	2
7	82	1	2
8	80	1	1
9	40	2	1
10	20	3	3
11	20	3	3

Table 1: Table comparing student pre-test scores with a human classification and the classification of the trained classifier

convolutional neural network architecture heavily inspired by work from Jaiswal and Nandi[30]. The emotion classification can be used to offer the student guidance. In the case of an extended period of negative emotions, the student might need to take a break from one section and try a different one, returning later to the original section.

Using the four components, the system evaluates student learning, providing support that is tailored to address the errors and misunderstandings that were made. The model can be further improved by gathering more student data. By analyzing student data from real classroom implementations of the system, common errors can be identified and the system can be updated to both better recognize more common errors and offer more helpful and specific support for those errors.

Model Evaluation

The random forest model was evaluated using quiz data from 12 students who previously participated in game testing. While the version of the game used in this test did not implement the random forest classifier, the game still shared a similar structure and gathered the same features with the exception of emotion data. To solve the missing feature, all student emotions were assumed to be neutral in this testing.

Table 1 compares a human expert score with the random forest classification. It also compares a pre-test score that was given to the students before they played the game. As shown, the classifier very reasonably represented both the pre-test score and the expert classification. Of note is student 4, who had high scores but long time and low confidence. Student 9 is also of note. Despite their low pre-test score, their in-game scores were very good, and are reflected in their classification.

Conclusion

This paper proposes the Personalized Instruction and Need-aware Gaming (PING) system. The system presents students with a problem to solve in a specific topic, in this case, digital logic design. While solving the problem and exploring the narrative game environment, students are offered individualized support to assist them in their learning process. To achieve the individualized support, a random forest classifier is used to determine the student's level of domain knowledge from information gathered as they play.

While the currently implemented system focuses on digital logic design, the system is general purpose and could be transferred to numerous different topics and game settings. The fully completed system should be able to appeal to students with all different learning styles and enhance learning for all students. By combining this system with a Problem-Based Learning approach, students should also gain more experience in applying their knowledge to solve complex, real-world problems.

References

[1] Xun Ge, Lourdes G. Planas, and Nelson Er. A cognitive support system to scaffold students' problem-based learning in a web-based learning environment. *Interdisciplinary Journal of Problem-Based Learning*, 4(1), 2010. doi: 10.7771/1541-5015.1093.

[2] John R. Savery. Overview of problem-based learning: De[U+FB01]nitions and distinctions. *Interdisciplinary Journal of Problem-Based Learning*, 1(1), 2006. doi: 10.7771/1541-5015.1002.

[3] John Hattie. Visible learning. 2008. doi: 10.4324/9780203887332.

[4] Aman Yadav, Dipendra Subedi, Mary A. Lundeberg, and Charles F. Bunting. Problem-based learning: Influence on students learning in an electrical engineering course. *Journal of Engineering Education*, 100(2): 253–280, 2011. doi: 10.1002/j.2168-9830.2011.tb00013.x.

[5] Robert Mayer. How engineers learn: a study of problem-based learning in the engineering classroom and implications for course design. doi: 10.31274/etd-180810-1693.

[6] APLUS+. Aplus+: Personalized learning, 2017. URL `https://theaplus.org/personalized-learning/`.

[7] Paul A. Kirschner, John Sweller, and Richard E. Clark. Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching. *Educational Psychologist*, 41(2):75–86, 2006. doi: 10.1207/s15326985ep4102_1.

[8] Harvey Siy, Brian Dorn, Carol Engelmann, Neal Grandgenett, Tracie Reding, Jong-Hoon Youn, and Qiuming Zhu. Sparcs: A personalized problem-based learning approach for developing successful computer science learning experiences in middle school. *2017 IEEE International Conference on Electro Information Technology (EIT)*, 2017. doi: 10.1109/eit.2017.8053436.

[9] Dermot Brabazon, Lynda Donovan, Mark Melia, Michael P. O'Mahony, Andrew Egan, and Barry Smyth. Supporting problem-based learning in moodle using personalised, context-specific learning episode generation. *Proceeding of 1st Moodle Research Conference*, 2012. doi: 10.1109/eit.2017.8053436.

[10] NAE. Nae grand challenges: Advance personalized learning, 2009. URL `http://www.engineeringchallenges.org/challenges/learning.aspx`.

[11] Diego Zapata-Rivera. Adaptive, assessment-based educational games. *Intelligent Tutoring Systems Lecture Notes in Computer Science*, page 435–437, 2010. doi: 10.1007/978-3-642-13437-1_95.

[12] Ying Tang, Kauser Jahan, and Talbot Bielefeldt. The effectiveness of an adaptive serious game for digital logic design. *2015 ASEE Annual Conference and Exposition Proceedings*, 2015. doi: 10.18260/p.24861.

[13] Ying Tang, Sachin Shetty, and Xiufang Chen. Educational effectiveness of virtual reality games promoting metacognition and problem-solving. *ASEE Annual Conference & Exposition Proceedings*, pages 25.495.1–25.495.7, 2012. URL `https://peer.asee.org/21253`.

[14] Ying Tang, Sachin Shetty, Talbot Bielefeldt, Kauser Jahan, John Henry, and Keith Hargrove. Sustain city - a cyberinfrastructure-enabled game system for science and engineering design. *The Journal of Computational Science Education*, 3(1):57–65, 2012. doi: 10.22369/issn.2153-4136/3/1/7.

[15] Christopher Franzwa, Ying Tang, Aaron Johnson, and Talbot Bielefeldt. Balancing fun and learning in a serious game design. *Gamification*, page 452–471, 2015. doi: 10.4018/978-1-4666-8200-9.ch022.

[16] Ying Tang, Chris Franzwa, and Aaron Johnson. Ci-team demonstration: Interactive and collaborative learning environment using virtual reality games promoting metacognition for science and engineering design in context, 2013. URL `http://sustaincity.rowan.edu/`.

[17] Ying Tang, Christopher Franzwa, Talbot Bielefeldt, Kauser Jahan, Marzieh S. Saeedi-Hosseiny, Nathan Lamb, and Shengtao Sun. Sustain city. *Design, Motivation, and Frameworks in Game-Based Learning Advances in Game-Based Learning*, page 57–91, 2017. doi: 10.4018/978-1-5225-6026-5.ch003.

[18] Robert Sottilare, Arthur Graesser, Xiangen Hu, and Keith Brawner. *Design Recommendations for Intelligent Tutoring Systems - Volume 3: Authoring Tools and Expert Modeling Techniques*. 06 2015. ISBN 978-0-9893923-7-2.

[19] Valerie J. Shute, Lubin Wang, Samuel Greiff, Weinan Zhao, and Gregory Moore. Measuring problem solving skills via stealth assessment in an engaging video game. *Computers in Human Behavior*, 63:106–117, 2016. doi: 10.1016/j.chb.2016.05.047.

[20] V.j. Shute and S. Rahimi. Review of computer-based assessment for learning in elementary and secondary education. *Journal of Computer Assisted Learning*, 33(1):1–19, 2017. doi: 10.1111/jcal.12172.

[21] Ying Tang, Sachin Shetty, and Xiufang Chen. Empowering students with engineering literacy and problem-solving through interactive virtual reality games. *2010 2nd International IEEE Consumer Electronics Societys Games Innovations Conference*, 2010. doi: 10.1109/icegic.2010.5716890.

[22] Barry Zimmerman. Self-regulated learning and academic achievement: An overview. *Educational Psychologist - EDUC PSYCHOL*, 25:3–17, 01 1990. doi: 10.1207/s15326985ep2501_2.

[23] Miguel Oliveira and J. Artur Serrano. Learning roadmap studio: new approaches and strategies for efficient learning and training processes. 01 2008.

[24] Trevor Hastie, Jerome Friedman, and Robert Tisbshirani. *The Elements of statistical learning: data mining, inference, and prediction*. Springer, 2017.

[25] Gérard Biau. Analysis of a random forests model, 2010.

[26] Robi Polikar. Ensemble learning. *Ensemble Machine Learning*, page 1–34, 2012. doi: 10.1007/978-1-4419-9326-7_1.

[27] Kuncˇeva Luˆdmila Ilieva. *Combining pattern classifiers: methods and algorithms*. John Wiley & Sons, 2014.

[28] Christine Chin. Classroom interaction in science: Teacher questioning and feedback to students' responses. *International Journal of Science Education*, 28(11):1315–1346, 2006. doi: 10.1080/09500690600621100.

[29] Bokyeong Kim, Hyungsung Park, and Youngkyun Baek. Not just fun, but serious strategies: Using meta-cognitive strategies in game-based learning. *Computers & Education*, 52(4):800–810, 2009. doi: 10.1016/j.compedu.2008.12.004.

[30] Shruti Jaiswal and G. C. Nandi. Robust real-time emotion detection system using cnn architecture. *Neural Computing and Applications*, 2019. doi: 10.1007/s00521-019-04564-4.