2020 16th IEEE International Conference on Automation Science and Engineering (CASE)
August 20-21, 2020, Online Zoom Meeting

Optimize Student Learning via Random Forest-Based Adaptive Narrative Game

Ryan Hare, *Member, IEEE*, Ying Tang, *Senior Member, IEEE*, Wei Cui, *Member, IEEE*, and Joleen Liang, *Member, IEEE*

Abstract— **This paper presents an adaptive narrative game system that focuses on sequential logic design. The system adapts a random forest machine learning model to estimate a student's current level of domain knowledge relative to the problem presented to him through his game-playing behavior data, such as time taken to find solutions, errors in solutions, and emotional indicators. Hints, prompts, and/or individualized lessons are then offered to the player to guide their learning in a positive and productive direction. Our preliminary pilot study demonstrates that the model can make accurate classifications, from which proper assistance can then be provided to individual students as they play.**

I. INTRODUCTION

The focus of education is to study the interplay of learning, human attributes, and social behaviors [1]. With recent technological advancement, this field of study has evolved into various new concepts, formats, and functions that support learning more effectively and efficiently. One of particular theme emerging is advanced personalized learning, which is the first of the 14 Grand Engineering Challenges set by The National Academy of Engineering for the 21st century [2]. The need, importance, and potential benefits to create a learning framework that tailors their assistance to a given student's individual knowledge and learning methods can be hardly overstated.

A personalized education system requires a learning environment for students to engage with. A narrative game would offer such an environment, in addition to providing players with both goal-based features, simulations of real-world problems, and goal-focused motivations. Narrative games that give students a problem related to their education can engage students in the problem-solving process and enhance their learning. These types of games have been strongly proven to both support student cognitive development [3], [4], [5], [6], [7], [8] and provide benefits in assessing students [9], [10], [11]. On top of the benefits of a narrative game, an Intelligent Tutoring System (ITS) [12] can be directly integrated into the game, providing a personalized system that can support and guide students along their respective learning path.

A vivid example of such integration can be found in the ITS developed by Squirrel AI [13], [14], where each student is recommended a unique learning path in the knowledge graph according to his own ability assessment in real time [15], [16]. It is noted that the games used in Squirrel AI are not connected with the ITS in the learning content wise, rather they serve as an incentive to students for their engagement.

To that end, a pre-established narrative game, called Gridlock [7], [17], [18], was transformed into a Personalized Instruction and Need-aware Gaming (PING) system. To determine a student's perceived level of domain knowledge, the PING system incorporates multi-component probing informed by Social Cognitive Learning Theory [19]. An updated random forest classifier is developed, where the level of support a student requires is determined from various features obtained from the student's game-playing behavior. In addition, the classifier is augmented with a feedback mechanism to iteratively improve its accuracy.

The proposed framework for this PING system combines cognitive psychology, statistical inference, machine learning techniques, education research, and sensor informatics to address the challenge of personalized engineering education. The rest of the paper is organized as follows. Section II gives the overview of Gridlock. Section III presents the random forest-based PING system. Section IV provides our preliminary assessment of the random forest classifier followed by the conclusions in Section V.

II. EXISTING GAME SYSTEM

Gridlock [7], [17] is a domain-specific game built in the Unity engine that focuses around educating students in digital logic design. As shown in Figure 1, the game begins with the player/student witnessing a traffic accident caused by a faulty traffic light. The student is then asked to redesign the logic controller for the traffic light. To nail down the design, the student must recognize that the traffic light controller represents a sequential circuit. The student can then create a finite state machine to complete the design.

Learning demands a learner strategically involved in the construction of meaning and be aware of his learning process [8]. To that end, three important metacognitive interventions,

This work was supported in part by the National Science Foundation under Grant 1913809 and Grant 1610164.

Ryan Hare is with the Department of Electrical and Computer Engineering, Rowan University, Glassboro, NJ 08028 USA harer6@students.rowan.edu

Ying Tang is with the Department of Electrical and Computer Engineering, Rowan University, Glassboro, NJ 08028 USA, and also with State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the Institute of Smart Education, Qingdao Academy of Intelligent Industries, Qingdao 266000, China, and the corresponding author. tang@rowan.edu

Wei Cui and Joleen Liang are with Squirrel AI Learning, Shanghai 200030, China.

Fig. 1. The street intersection and main narrative structure of the game [18].

Roadmap, Know-Want-to-know-Solve (KWS), and Think-Aloud-Share-Solve (TA2S), are carefully designed into Gridlock to offer students necessary guidance. Roadmap endows students with the capability to locate relevant information and key concepts within study materials. KWS evokes students' thinking of thinking, activates their prior knowledge, reviews what they have done, and then motivates them to seek more knowledge to improve their solution. TA2S is a variation of collaborative learning strategies that promotes social stimulation to learning. More details about these interventions can be found in [8].

Gridlock has been demonstrated as a good learning tool through an evaluation conducted in seven courses with over 300 students at Rowan and Tennessee State Universities [8]. This prior study also surveyed students on their views of the learning support provided. Some felt the current support provided necessary assistance in identifying domain knowledge. Others felt that the expert guidance could be more detailed with additional coaching. These responses demonstrated that students have unique and individual ways of learning and that Gridlock did not achieve its full potential due to lack of personalized feedback and guidance. If the game system can better extract personalized differences and provide a wider range of appropriate customized support, the system will be significantly more effective and efficient. Tackling this challenge is the goal of the PING system, described in section III.

III. Personalized Instruction and Need-aware Gaming (PING) System

The key step of augmenting Gridlock to better offer individualized learning support is to establish a mechanism of creating and transferring learner profiles. This mechanism must underpin assessment in game and allow mapping of those learner profiles to differentiated coaching. Such profiles are not immediately available at the initialization of the game, and can dynamically change as the game progresses. The problem of dealing with the lack of data necessary for decision-making falls under the broader area of system identification. A common approach is to measure the behavior of the system and the external influences with the aim to determine mathematical relations. This idea is adopted to leverage Gridlock into the PING system using an updated Random-Forest-based feedback loop control as shown in Figure 2. The end result is a system composed of four main components: Social-Cognitive-Theory-based probing, a Student Knowledge Database, an Instruction Database, and the Student Comprehension Model.

A. Social-Cognitive-Theory-based Probing

To better characterize learner profiles, multi-dimensional measurement grounded in SCLT is carried out with the attempt to replicate traditional features that educators would look for in a classroom setting. The four main components include: Probes, Time, Errors, and Emotion.

1) Probes: Question prompting has been proved to be an effective instructional strategy for directing learners to the most important aspects of a problem, as well as encouraging self-regulation, self-reflection, and evaluation [20]. Throughout the PING system, there are quizzes given at certain points with sections to gauge student knowledge and measure its evolution during the game. In addition, the question prompts within the game have also been augmented with a self-rating system where students rate their confidence in their answer. Both the question scores and confidence ratings are used for classification.

2) Time: When students take significantly longer than average to respond to a component of the problem, it indicates that they either lack time management and focus or are running into technical difficulties. As the student plays through the various sections of the game, time measurements for problem-solving steps are constantly recorded.

3) Errors: In addition to students' answers to question prompts, the PING system diagnoses student designs in Verilog code and provides opportunities for them to repeat 'task-performance-feedback cycle' through resubmission.

4) Emotions: The PING system makes use of webcams to constantly track fluctuations in the emotions of a student through his facial expressions as learning progresses. A pre-trained emotion classification system utilizing a convolutional neural network architecture [21], [22], [23], [24] is adopted. From the CNN, the system obtains a classification of one of seven emotions, Neutral, Happiness, Sadness, Disgust, Fear, Surprise, and Anger. The system also tracks emotion through measuring irregular mouse movements and key presses.

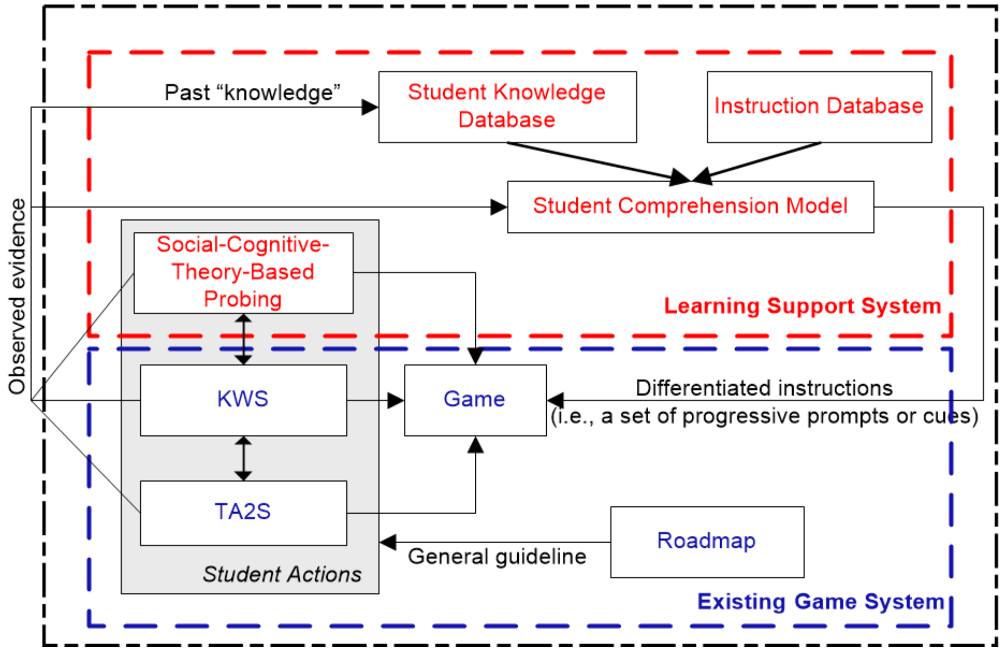

Fig. 2. Architecture of the PING system. Components enclosed in blue represent components already within the existing game system. Components in red represent the PING system components to be implemented in the game.

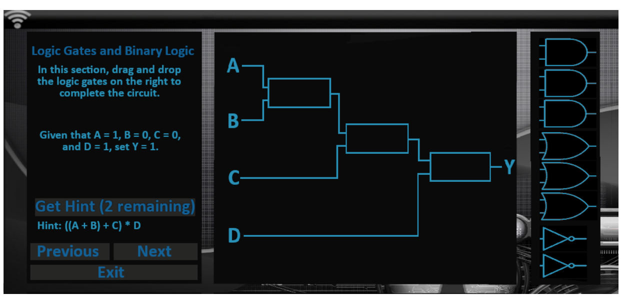

Fig. 3. Example of a concept-specific section within the game focusing on logic gates and binary logic. In this case, students are tasked with placing the correct logic gate in the empty slot to complete the circuit.

B. Student Knowledge Database

Instead of tracking student behavior along problem solving paths, the PING system attempts to determine if the student grasps all of the information necessary to solve the problem. To that end, the entire game is divided into several sections, each of which corresponds to a subset of the knowledge space for the problem-solving in the game. Figure 3 shows one of these concept-specific sections. Within each of the game sections, the Student Knowledge Database logs events and records of the student behavior, including responses to questions, mouse-click history, facial emotion signals, key-press patterns, etc. This data forms a set of features for classification.

Let us consider a set of measurement data of a student $x_i \in \mathcal{R}^r$. Each $x_i = \{f_j, j = 1, 2, ..., r\}$ consists of r "features" and is associated to a label $y_i \in \mathcal{R}^c$ that permits one to distinguish between c classes. For our case, the following features are used that correspond to the signals derived from the SCLT-based probes. A set of training data is then used to identify the decision rules for the classification in the Student Comprehension Model to accurately categorize any new student x into three classes, $y_i \in \{1, 2, 3\}$.

- Section number as an integer between [1, 7] to show which knowledge section the current data pertains to.
- Prior section classification between [1, 3]. For the first section, this is assumed to be 1.
- Score for each question as a real number between [0.0, 1.0]. Correct answers are always assigned 1.0, while incorrect answers are assigned a value between 0.0 and 0.9 depending on how close they are to the correct answer. The questions are multiple-choice questions, and are all pulled from a pool of questions programmed into the game. When the question pool is created, the question creator subjectively determines how "correct" each answer is by assigning the value for incorrect answers.
- Time for each question as a real number measuring the total duration in seconds taken to submit an answer.
- Confidence for each question as an integer between [1, 5] showing at what level the student rated their own confidence in their answer. 1 is "not confident at all" and 5 is "very confident".
- Emotion Percentages as 7 different real numbers each between [0.0, 1.0] showing the percentage of updates over the course of the section where each emotion was present.
- Most Prevalent Emotion as a string for the most present emotion over the course of the section, for example "Happiness".
- Frustration as a real number between [0.0, 1.0] mea-

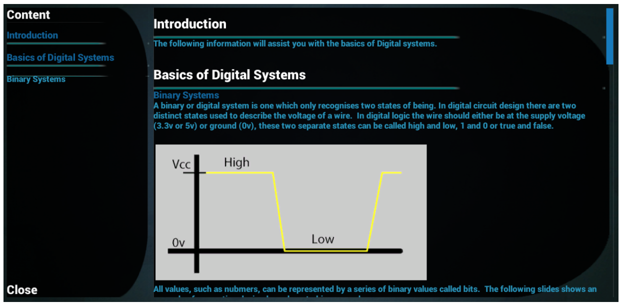

Fig. 4. Excerpt from the study guide materials on the basics of digital systems.

sured over the course of the section. Frustration in this case is defined as a function of the player's total mouse movement per second in pixels (M) and total number of key presses per second (K) as shown in Equation 1. The threshold values (19 for K and 6450 for M) are determined through an experimental study of measuring the total number of key presses and mouse movements made in one second for average people.

- Frustration Frames as an integer which shows the number of updates over the course of the section where the frustration score exceeded a threshold, in our case 0.5.

$$F = Max(\frac{Min(K, 19)}{19}, \frac{Min(M, 6450)}{6450}) \tag{1}$$

C. Instruction Database

The Instruction Database contains a set of hints, prompt questions, study materials, and pop-up statements that the system can give to the student. The database is divided into subsets that pertain to the content-specific sections of the game. As the student plays, the Student Comprehension Model pulls instructions from the database based on the student's classification. In the current game iteration, instructions are divided into basic and advanced instructions with the advanced instructions providing more in-depth instruction and guidance. Students who are classified as a 3 are shown advanced instructions while students who are classified as 2 are shown basic instructions. Figure 4 shows an example of some of the content that would be shown to a struggling student.

D. Student Comprehension Model

The PING system inherently deals with uncertainty within human behavior. The goal is to correctly classify a student's level of domain knowledge and, based on those predictions, offer the student personalized learning support to direct their learning in a positive and productive direction. The Student Comprehension Model (SCM) is the main decision-making portion of the PING system for student classification. The predictions made by the SCM must be accurate enough to correctly determine what help the student needs while also being general enough to work well for each unique student that plays the game.

Our prior study proposed a k-Nearest-Neighbors (kNN) classifier [3], [18]. However, kNN is not suitable for handing categorical data (for example, emotion measurements) as kNN uses a distance measure to classify data. In addition, larger data sets can cause greatly increased computation time and innaccurate classifications when using a kNN classifier. To overcome these shortcomings, this paper adapts a random forest classifier, which is an ensemble of decision trees. The decision tree basis of this classifier means that it can easily handle categorical, ordered, and numerical data and be less of a "black box" than other classifiers, such as support vector machines or neural networks. Compared to a single decision tree classifier, the random forest is both more stable and resistant to errors and variations in the data [25], [26], and has improved performance [27], [28]. The relatively low computational cost of training is another advantage of the random forest.

As defined in Section III-B, a set of training data $T = \{(x_i, y_i), i = 1, 2, .., n\}$ is first needed for the initial random forest. For our case, the data is collected from the development of Gridlock and annotated by a human expert and used as one part of the training data. Meanwhile, according to the feature relations described in Section III-B, another portion of the training data is populated using multivariate normal distributions for each class. With number of decision trees and maximum tree depth predefined, the random forest training algorithm [29], [30], **RFTrain**, is then executed on the training data to create the initial random forest RF_0. In this paper, the number of decision trees and the tree depth are set as 100 and 5, respectively.

While each new student x_i plays the game, their learning profile is reshaped at each game section i through the use of subject-specific tests, quizzes, or mini games, and modeled as a feature vector. Inputting the feature vector to the current random forest, the classification algorithm [30], **RFClassify**, is applied to associate x_i with label y_i. Based on the labeled data (x_i, y_i), support is pulled from the Instruction Database that is tailored directly to the student's real-time learning behavior. As the game progresses, the system continues to observe the student until the submission of their final solution.

Given that the ultimate learning goal is to design the traffic light controller, our contention is that the efficiency and effectiveness of the student's solution reflects their true learning performance. To that end, the system records the number of errors in the solution, the type of errors, and the number of attempts at submission, based on which a true performance label $\hat{y}$ is given. The data $(x, \hat{y})$ is then fed into the training set to improve the training accuracy for future students. The progressive random forest classification process is shown in Algorithm 1.

IV. Experimental Study

To verify the classification accuracy of the random forest classifier, a recent pilot of the game had students in a course at Rowan University play through the game. Students who played the game were in a more advanced class, playing the game as a review of previous content. As such, the classifications were compared to their final grades in the course relevant to the content of the game. Table I shows the average classification over all sections of the game assigned to 7 different students who participated in the study. For this test, the system was trained on a mix of generated data and expert-labeled data. This focus group shows that the system is able to classify students in a way that correlates with the proficiency they demonstrated in a related course.

More student data was gathered from 19 students as they played through the game for the first time with no prior knowledge. Students' grades in the course relevant to the content of the game were used as a ground truth for the final classification. Grades from A to B+ were mapped to 1, B to C+ were mapped to 2, and anything lower was mapped to 3.

To prove that the system can evolve with added data, a subset of 10 students was initially classified using a small pool of expert-labeled data and a large pool of generated data. These 10 students were classified with an 88.57% accuracy when compared to their final grades in the relevant course. The data from those 10 students was then fed into the training set. Using the updated training data set, all 19 students were classified. The final accuracy over all 19 students was shown as 91.73%.

Algorithm 1 Random Forest Classification

Input:

Labeled data set T;

Random Forest $RF = \varnothing$;

Unlabeled data set $X = \{x_i, i = 1, 2, ...\}$;

Random forest training algorithm with relevant parameters - **RFTrain**;

Random forest classification algorithm - **RFClassify**;

Output:

The set $Y = \{y_i, i = 1, 2, ...\}$ where y_i is the label for x_i

for i = 1, 2, ... **do**
    Call **RFTrain** with T and relevant parameters to obtain the random forest.
    Update RF
    for j = 1, 2, ... **do**
        if j = 1 **then**
            Set the prior classification as 1, **update** x_i
            Call **RFClassify** with RF to obtain classification y_i
        else
            Set y_{j-1} as prior classification, **update** x_i
            Call **RFClassify** with RF to obtain classification y_i
        end if
    end for
    Observe and feedback true label of x to $T = T \cup (x_i, \hat{y}_i)$;
end for

The accuracy increase shows that updating the training data set with real data does improve the classification accuracy of the system relative to final grades in a relevant course, which represents a significant measure of student domain knowledge.

TABLE I

TABLE COMPARING STUDENT GRADES IN A RELATED COURSE TO AVERAGE CLASSIFICATIONS WITHIN THE GAME.

Student Grade	Student Level	Average Classification
A	High	1.00
A	High	1.14
A-	High	1.14
B+	High	1.29
B	Medium	1.71
B-	Medium	1.86
C-	Low	2.14

V. CONCLUSION

This paper proposes an adaptive learning game system where students receive individualized support to enhance their learning of a topic; In this case, digital logic design. An updated random forest classifier is developed, not only to classify student learning profiles as they play, but to iteratively improve training accuracy through a feedback mechanism. The classifier is shown via a focus group test to achieve an accuracy of 88.57% using a small pool of real, expert-labeled data supplemented with generated data, but the accuracy was increased to 91.73% when a small amount of real-world data was labeled with true student observations and fed back into the training set. Future work will focus on continuing to test the system and to gather additional real data, as well as comparative studies to show the effect of using the system versus not using the system. As additional data is collected and additional observations are made to establish classifications, the system can also be retrained for increased performance.

REFERENCES

[1] F.-Y. Wang, Y. Tang, X. Liu, and Y. Yuan, "Social education: Opportunities and challenges in cyber-physical-social space," *IEEE Transactions on Computational Social Systems*, vol. 6, pp. 191–196, 2019.

[2] NAE, "Nae grand challenges: Advance personalized learning," 2009. [Online]. Available: http://www.engineeringchallenges.org/challenges/learning.aspx

[3] Y. Tang, K. Jahan, and T. Bielefeldt, "The effectiveness of an adaptive serious game for digital logic design," *2015 ASEE Annual Conference and Exposition Proceedings*, 2015.

[4] Y. Tang, S. Shetty, and X. Chen, "Educational effectiveness of virtual reality games promoting metacognition and problem-solving," *ASEE Annual Conference Exposition Proceedings*, pp. 25.495.1–25.495.7, 2012. [Online]. Available: https://peer.asee.org/21253

[5] Y. Tang, S. Shetty, T. Bielefeldt, K. Jahan, J. Henry, and K. Hargrove, "Sustain city - a cyberinfrastructure-enabled game system for science and engineering design," *The Journal of Computational Science Education*, vol. 3, no. 1, p. 57–65, 2012.

[6] C. Franzwa, Y. Tang, A. Johnson, and T. Bielefeldt, "Balancing fun and learning in a serious game design," *Gamification*, p. 452–471, 2015.

[7] Y. Tang, C. Franzwa, and A. Johnson, "Ci-team demonstration: Interactive and collaborative learning environment using virtual reality games promoting metacognition for science and engineering design in context," 2013. [Online]. Available: http://sustaincity.rowan.edu/

[8] Y. Tang, C. Franzwa, T. Bielefeldt, K. Jahan, M. S. Saeedi-Hosseiny, N. Lamb, and S. Sun, "Sustain city - effective serious game design in promoting science and engineering education," *Design, Motivation, and Frameworks in Game-Based Learning, IGI Global*, p. 57–91, 2017.

[9] R. Sottilare, A. Graesser, X. Hu, and K. Brawner, *Design Recommendations for Intelligent Tutoring Systems - Volume 3: Authoring Tools and Expert Modeling Techniques*, 06 2015.

[10] V. J. Shute, L. Wang, S. Greiff, W. Zhao, and G. Moore, "Measuring problem solving skills via stealth assessment in an engaging video game," *Computers in Human Behavior*, vol. 63, p. 106–117, 2016.

[11] V. Shute and S. Rahimi, "Review of computer-based assessment for learning in elementary and secondary education," *Journal of Computer Assisted Learning*, vol. 33, no. 1, p. 1–19, 2017.

[12] D. Zapata-Rivera, "Adaptive, assessment-based educational games," *Intelligent Tutoring Systems Lecture Notes in Computer Science*, p. 435–437, 2010.

[13] Z. X. Z. Z. H. Li, W. Cui and M. Feng, "Yixue adaptive learning system and its promise on improving student learning," *Proceedings of the 10th International Conference on Computer Supported Education*, 2018.

[14] W. C. M. Feng and S. Wang, "Adaptive learning goes to china," *Artificial Intelligence in Education*, vol. 10948, pp. 89–93, 2018.

[15] K.-P. T. W. Cui, Z. Xue, "Performance comparison of an ai-based adaptive learning system in china," *Chinese Automation Congress, Xi'an: IEEE*, 2018.

[16] J. S. G. S. J. L. W. Cui, Z. Xue, "The item response theory model for an ai-based adaptive learning system," *18th International Conference on Information Technology Based Higher Education and Training, Magdeburg: IEEE*, 2019.

[17] Y. Tang, S. Shetty, and X. 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.

[18] Y. Tang, J. Liang, R. Hare, and F.-Y. Wang, "A personalized learning system for parallel intelligent education," *IEEE Transactions on Computational Social Systems*.

[19] B. Zimmerman, "Self-regulated learning and academic achievement: An overview," *Educational Psychologist - EDUC PSYCHOL*, vol. 25, pp. 3–17, 01 1990.

[20] X. Ge, L. G. Planas, and N. Er, "A cognitive support system to scaffold students' problem-based learning in a web-based learning environment," *Interdisciplinary Journal of Problem-Based Learning*, vol. 4: Iss. 1, p. Article 4, 2010.

[21] S. Jaiswal and G. C. Nandi, "Robust real-time emotion detection system using cnn architecture," *Neural Computing and Applications*, 2019.

[22] Y. Miyakoshi and S. Kato, "Facial emotion detection considering partial occlusion of face using bayesian network," in *2011 IEEE Symposium on Computers Informatics*. IEEE, 2011, pp. 96–101.

[23] S. Li and W. Deng, "Deep facial expression recognition: A survey," *arXiv.org*, 2018. [Online]. Available: http://search.proquest.com/docview/2072077427/

[24] "Facial expression recognition with data augmentation and compact feature learning," in *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018, pp. 1957–1961.

[25] T. Hastie, J. Friedman, and R. Tisbshirani, *The Elements of statistical learning: data mining, inference, and prediction*. Springer, 2017.

[26] G. Biau, "Analysis of a random forests model," *Journal of Machine Learning Research*, vol. 13, pp. 1063–1095, 2012.

[27] R. Polikar, "Ensemble learning," *Ensemble Machine Learning*, p. 1–34, 2012.

[28] K. L. Ilieva, *Combining pattern classifiers: methods and algorithms*. John Wiley Sons, 2014.

[29] APLUS+, "Aplus+: Personalized learning," 2017. [Online]. Available: https://theaplus.org/personalized-learning/

[30] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.