Design and Implementation of an Activity-Based Introductory Computer Science Course (CS1) with Periodic Reflections Validated by Learning Analytics

Mohsen Dorodchi[1], Aileen Benedict[1], Devansh Desai[1], Mohammad Javad Mahzoon[2], Stephen MacNeil[1], Nasrin Dehbozorgi[1]
Departments of [1]Computer Science, and [2]Software and Information Systems
University of North Carolina at Charlotte
Charlotte, USA
(Mohsen.Dorodchi, abenedi3, ddesai14, mmahzoon, smacnei2, ndehbozo)@uncc.edu

***Abstract*— This research to practice full paper provides preliminary evidence that integrating reflections is a significant feature to identify at-risk students early in a semester as verified and validated by a sequence-based learning analytics model. We've devised an active learning classroom model which incorporates reflection at multiple points in students' learning experience. This active learning model adopts Kolb's Learning model to provide a coherent and connected set of activities before, during, and after the class. Unlike periodic assessment through testing, reflections can provide nearly-real-time information about student's experiences in class. We extract sentiment feature vectors to capture students' affect from written reflections. These features typically aren't assessed on tests or during in-class activities. These features were extracted automatically using LIWC (Linguistic Inquiry and Word Count) is a tool for applied natural language processing) which is less cumbersome to implement than manually reading the written reflections. We find that using these sentiment feature vectors extracted from the reflections in our learning model increased accuracy while decreasing time-to-detect at-risk students significantly.**

Keywords—Active Learning, Reflection

I. Introduction

Many courses are designed with students' learning styles in mind [15]. The challenge associated with designing for these students' learning styles is considering the diversity of learning styles while still making sure that students are learning effectively. In other words, how can we design and evaluate a course based on evidence rather than "intuition", since such intuition may not always be relevant [13]. The evidence-based model in education, then, is a logical approach to make sustainable improvements in pedagogy. This practice is defined as "the integration of professional wisdom with the best available empirical evidence in making decisions about how to deliver instruction" [41].

The first programming course of computer science, aka CS1, is known as a problematic course due to many factors such as diverse student backgrounds, large class sizes, lecture-based auditorium-style teaching, etc. Therefore, it is important for informed decisions to be made early on about the course model [43]. In this work, we are proposing a course model based on students' learnings styles, which includes monitoring mechanisms by integrating reflections applied to a 100+ student class size. By adopting active-learning ideas, the course includes many unique and diverse types of activities. Furthermore, we use these reflections and course grades to identify students that are at-risk of DFW in the course. Based on the risk classification, we are able to then update our pedagogy in subsequent courses or intervene to help specific students earlier in the semester. Through this process, we can provide opportunities for student reflection, identify at-risk students, improve our course design, and intervene when necessary. Our learning analytic model integrates sentiment feature vectors which captures student affect as it relates to course material and design. It is based on a sequence data model which accounts for temporality and can therefore be used to make classifications at multiple points in time.

We focus on the following research questions regarding the learning model as it relates to our course design:

RQ1. Does including student reflections as a feature in the learning model help predict at-risk students?

RQ2. Does including student reflections as a feature in the learning model enable earlier predictions?

RQ3. What is the effectiveness of infusing student reflective practices between activities and throughout the course?

To answer these questions, we have used a sequence-based learner analytic model which incorporates reflections as a feature for classifying at-risk students. We will discuss how we have integrated reflections into our course and how these reflections have been used as a feature in our learner analytic model. We compare the effect of integrating reflections by comparing them to other common performance-related features such as tests, assignments, and activities.

II. Theoretical Framework

A. Course Model

The teaching paradigm for engineering and computing disciplines is shifting from traditional lecture-based teaching methods to designing learning experiences, processes, and environments [10]. According to Fink, an integrated course design starts from analyzing the "situational factors", followed by formulating the "learning goals" as well as designing the "feedback and assessment procedures." The design and selection of the teaching/learning activities fulfills this process [11]. Activity-based active learning seems to be one of the desirable delivery methods for such teaching and learning activities providing excitement and fun while emphasizing on

978-1-5386-1174-6/18/$31.00 ©2018 IEEE

learning [3, 9, 32, 43]. Considering the above factors, a coherent course would have a complete alignment between the activities, assessment, and learning goals and outcomes. In addition, the need for an educational measurement of student knowledge aligned with activities and learning goals, which is beyond traditional tests, as well as the methods to make inferences about student learning are instructionally essential in this model of course design [28].

B. *Reflection*

Reflection is described as a deliberate process or "activity in which people recapture their experience, think about it, mull it over and evaluate it" [5]. It is through this process of evaluation that students generalize knowledge from concrete experiences. This perspective is rooted in Kolb's model for experiential learning [18,19]. Experiential learning relies on the constructivist view that knowledge is constructed by the learner [40]. This view, established by Piaget [30], defines learning as an active process that is subjective to each student. Students synthesize new information and experiences with their existing understanding of the world. From this perspective, reflection is an important aspect of learning because it is one of the places where this synthesis happens.

Self-assessment, which is a form of reflection, is a technique for having students evaluate their performance and behaviors [23]. Self-assessment is often used in educational settings because it scales well as a way to assign grades to students for large classes. Self-assessment, has been shown to be relatively consistent with grades given by teaching staff [4]. It is however noted by David Boud that weaker students have a tendency to overestimate their abilities, while stronger students have a tendency to slightly underestimate their abilities. Self-assessment is often used in conjunction with peer assessment [38] to help triangulate students' performance [8]. Combining peer and self-assessment has been scaled to large-scale courses as well [20].

Reflection has been incorporated in engineering and computing courses for variety of purposes. For example, reflection has been used in computer science as a way for instructors to understand the experiences of students to improve their pedagogy [35]. Based on a systematic literature review by Sepp et al. the number of engineering education papers that reference reflection have increased significantly over time [34]. Turns et al. have created a framework for considering the multiple aspects of reflection when implementing it in engineering courses [39]. All of this attention to integrating reflection in both CS and engineering in general is encouraging for analyzing potential opportunities of leveraging reflection for purposes such as learner analytics.

In addition to this widespread use and the aforementioned benefits, there are also a number of challenges associated with analyzing reflections but a primary problem is that the "... learning outcomes of reflections are subjective knowledge rather than objective knowledge. Only the person doing the reflection can assess whether learning has occurred that is significant to them" [6]. Similarly, students frame their reflections to please the person that is doing the assessing, which results in less authentic and valuable insights for the student. This is potential for bias is an example of the Hawthorne effect [12]. For this reason, designing and assessing reflective practices needs to be done very carefully. In this paper, we use reflections to help with our risk classification model and provide participation points (or extra credit for non-required reflections) to ensure that students complete the reflections. Students are not told that their reflections will be used for classification purposes to avoid biasing their reflections.

C. *Learning Analytics*

To verify the effectiveness of our designed course model, learning analytics methods are applied. Learning analytics is an emerging discipline within data science/analytics, concerned with developing methods for exploring the unique and increasingly large-scale data that are collected from educational settings. The analysis and visualization are to understand and optimize students' learnings and the environments in which it occurs. These methods are developed and applied the same way as in general data analytics exploiting statistical and machine-learning for prediction, clustering, outlier detection, discovery with models, text mining, knowledge tracing, relationship mining, etc. to search for unobserved patterns and underlying information in learning processes [1].

Learning analytics have been used in many situations. For example, Yuliya et al. explores which student difficulties arise within CS1 courses by mining data from CodeLab – a "web-based interactive programming problem system" – finding that conditions and loops are the main challenges for students [7]. In conclusion, they also further encourage the use of large data from many institutions to gather greater insight. Agudo-Peregrina et al. have applied learning analytics, specifically bivariate correlation analysis, to find the correlation between interactions (i.e., student to student interactions within LMS, student interactions with LMS (learning management system) content, and student interactions with the professor) and student performance. In another study, learning analytics was used to identify significant behavioral indicators of learning. Results showed that students' regular study, times of assignment submission, number of login sessions, and proof of reading course information were all significant factors in predicting course achievement [44].

III. Related Work

A. *Course Model*

As discussed earlier, Kolb's experiential learning theory provides a solid model for teaching courses requiring multi-stage activities. Kolb's experiential learning cycle has been used previously in various courses to structure class activities and content delivery [15, 33], and are useful to "move students quickly through multiple levels of learning" [33]. In a CS2 course, instruction typically begins with a concrete experience and an explanation of "why" the experience is important [15]. Following this, they tell students "what" they will need to know and then "how" to apply the material to real problems. Finally, students actively experiment with other alternatives and different possibilities. This systematic process of introducing and scaffolding the delivery of material was beneficial in their course. Furthermore, repeating this cycle each week helped students develop a consistent understanding of the material. As

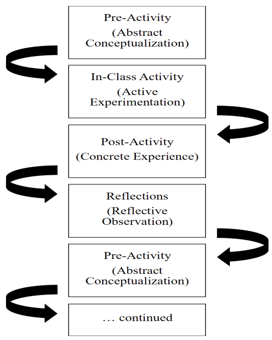

Fig. 1. The iterative learning model based on Kolb's Learning Model [18,19].

argued by Harb and Terry, as well as by Howard et al., teaching in this way meets the many needs of students [14].

We have previously proposed pre/in/post activities based on Kolb's model [9]. This course design led to more overall A's and B's and less C's and D's. The number of F's and W's did not change significantly which could be interpreted that if students are engaged in activities, they experience more success [9]. In Fig. 1., a representation of Kolb's model is presented.

B. Reflection

Turns introduces a framework on reflections and its different elements (Turns, 2014). She relates the reflective practices to the metacognitive concepts of self-awareness and self-assessment as issues being "foregrounded". As metacognition is defined as knowing about knowing [24], reflections should provide such opportunity for students to grow their metacognitive skills.

In a similar study, reflections are used before and after tests to measure students level of confidence [27]. By incorporating metacognition, the goal was to encourage students to actively think about their own level of understanding so that adjustments could be made.

Stone also designed weekly reflections where students responded to four identical prompts every week asking about what they learned, what they enjoyed, what was challenging or frustrating, and what their current confidence levels were [35]. One of the mentioned study challenges was the difficulty in the qualitative analysis of the student responses. Instead of manually analyzing the student reflections, we chose to use a tool called Linguistic Inquiry and Word Count (LIWC) to aid in the analysis, which we discuss further in *Data Collection and Analysis*.

C. Learning Analytics

Many features can be analyzed to classify student performance. These include aspects such as study patterns, exhibited emotions, and temporal features. Through analyzing these features, it is possible to identify at-risk students to improve the course or to intervene on their behalf.

In a study of 530 college students, a learning analytics model was used to predict course achievement as measured by their activities inside a learning management system (LMS) [44]. This demonstrated that their pattern of study, late submissions, and whether they reviewed the materials was predictive of performance. In another study, students' emotional reactions were correlated with student performance on programming assignments [21]. This work influences our use of affect to identify risk.

In addition to these features, temporal aspects can also be analyzed. However, based on multiple surveys [25, 29, 36], we identified lack of attention to the temporal aspects of the student data. Methods reviewed in the surveys used statistics or machine learning techniques operating on a feature vector representation of each student having non-temporal features such as demographic information, student academic information, course grades, and learning management system (LMS) logs in aggregated form. Some authors have argued that temporal aspects of student data deserve more attention, and temporal analysis yields a paradigm shift addressing new research questions in learning analytics [26]. Similarly, previous works in computer supported collaborative learning (CSCL) and self-regulated learning (SRL) emphasizes on the importance of temporal features in student data [16, 31, 2].

Wolff et al. identified at-risk students using a decision tree model [42]. This model is also temporal and includes assignment scores and number of clicks in the virtual learning environment (VLE) across specific time periods to predict future performance drops for students who were performing well. They acknowledged the fact that the number of clicks cannot predict successful behavior saying "There were students who clicked a lot and still failed, or those who clicked hardly (if) at all and yet passed" [42]. They created time frames for counting the number of clicks to break down the general feature of "number of clicks" into "number of clicks in a time window". This is similar to the concept of "nodes" in our sequential model (see section *Data Collection and Analysis*). Our model also groups heterogeneous data together into nodes. Each node represents a group of activities, tests, assignments, or reflections during a time frame. Grouping this heterogeneous data together constitutes a specific node. We also consider the dependency that may exist between nodes.

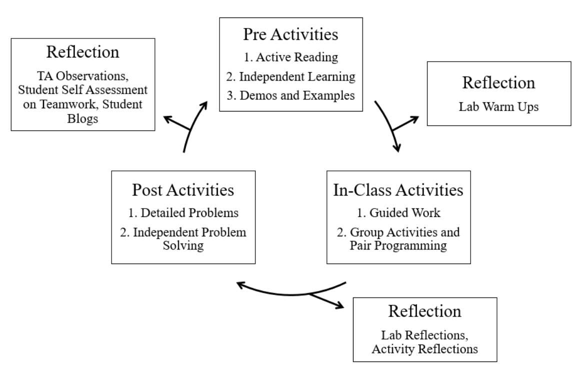

Fig. 2. Our course model with integrated reflections.

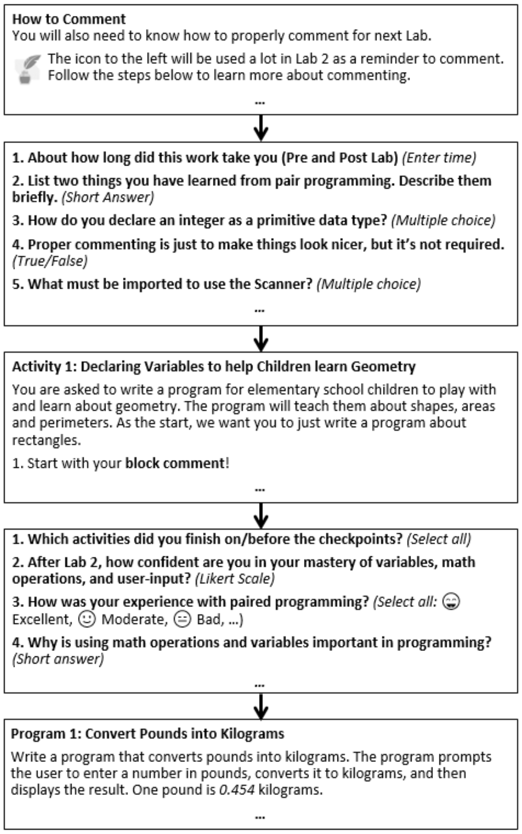

Fig. 3. Illustrating the flow of one week's Lab cycle, taking sample instructions and questions from a pre-activity, reflection, in-class activity, reflection, and post activity.

IV. Course Model

In our course model, we have adapted Kolb's experiential learning cycle by removing and replicating the reflection stage multiple times throughout the cycle, as shown in Fig. 2. Kolb's model has four stages while our resulting model has three distinct stages with reflection integrated throughout, rather than occurring only once each cycle. Students plan for the upcoming week with prep work assignments, have active learning experiences in lab, and then extend their learning on assignments at home. By integrating reflection throughout, our model becomes a 3-stage process: prep work that helps students prepare for upcoming classes, in-class activities to further learn and experiment on the material, and post-class assignments that extend concepts from the course. We then have students reflect on each of these aspects at multiple times throughout the course.

The activities are designed with the goal of both challenging the students while encouraging them to enjoy the group and social aspects of term activities [9]. When designing activities for each phase and creating tests for assessing students' learning, Bloom's Taxonomy was also applied.

An example of one week's iteration is shown in Fig. 3. First, the students complete the pre-lab homework (a type of *pre-activity*, as shown in the cycle in Fig. 2) meant to prepare them

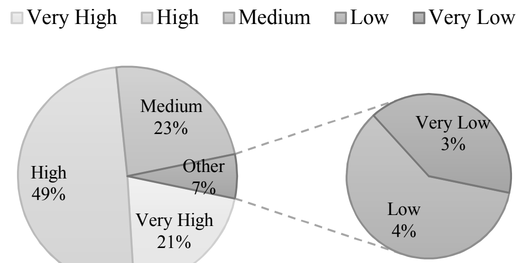

Fig. 4 Students' perception about how much they learned Java programming

before coming to the lab. Then, at the beginning of lab, a lab warm up is completed (a type of *reflection*) which asks questions relating to their preparation work. The lab itself is a type of *in-class activity*, where students complete programming activities to strengthen their knowledge of what was learned that week. After lab, students complete the lab reflection (another type of *reflection*), responding to questions about the lab itself and their own learning experiences. Finally, students complete a post lab (a type of *post activity*) independently for homework to further review what they have learned.

Many of our own reflective prompts were aimed at having students think about their own learning experiences, hence the metacognitive attributes can be included in our data. Fig. 4 shows an example of a question from one of the reflections which asks the students near the end of the course how much they thought they learned about Java programming. Similarly, we also wanted to encourage students to think about their progress and level of understanding throughout the course. Some of our reflective prompts were designed in this way. However, we did not only include reflections just a few times throughout the semester. Instead, it was an iterative process that occurred much more frequently, such as at the end of many in-class activities. Furthermore, we did not limit our prompts to only students' attitude toward their learning experiences but asked a mixture of questions regarding many different aspects of their learning experiences.

V. Data Collection and Analysis

The data was collected from 91 university students enrolled in the Introduction to Computer Science (CS1) course in the Spring semester of 2017. The course demographics consisted of 21.9% female students and 72.5% CS-majors. The data has three main categories per student:

1. Student background information
2. Student performance scores
3. Student reflections and self-assessments

Each of these three main categories includes specific attributes which are used in our algorithm as features. More specifically, student background includes attributes such as age group, gender, and major. Student performance scores include grades for all quizzes (18 total), pre/post labs (16), long assignments (4), lecture tests (4 total, including three during the semester

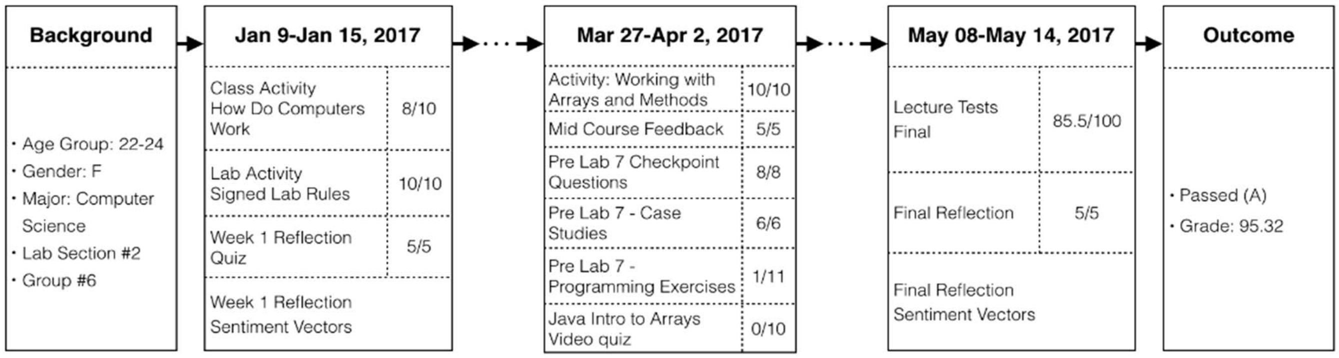

Fig. 5. A sample student's sequence.

and one final), lab tests (4), lab/lecture activities (37), and extra credit activities (4) for a total of 87 different columns per student. Reflections are informal surveys taken by students regularly after class (lab/lecture) activities, assignments, quizzes, and tests for a total of 23 reflections per student. Students reflected on their learning of different course topics, as well as on the learning processes, group activities, or the tests/assignments. Some of the reflections, therefore, were mandatory as a part of the activity, while others were optional extra credit activities. We ended up with a heterogeneous dataset for different reasons: 1) we have both numerical and textual data; 2) data items' frequency of occurrences are different such as weekly, biweekly, or monthly; and 3) the data included objective and subjective measures, as well as self or group assessment by students.

All the 110 different grades and quantified reflections are spread over in our dataset based on the date of the activity. This highlights their strong temporal dependencies with each other. Therefore, this data is a good candidate for using temporal data analysis models. It should be emphasized that the temporal dependency of the data items comes from the fact that students must do different types of activities in the lab and lecture as explained while providing reflections over time. The activities are all dependent and build on top of each other. In addition, students were reflecting on their learning and outcomes of activities which suggest the strong dependency as shown in Fig. 5. In other words, it is possible that a student who received low grades for the first few weeks of the course might change their study pattern to make up for the low performance. Consequently, we have dependencies in activities themselves as well as dependencies in the time between reflections and activities.

A. Proposed Data Model

Our goal was to discover the trends of students' activities throughout the semester, predict the outcome (Success or Fail) and discover the impact of reflections on the prediction. To do so, we built a temporal data model called the *student sequence model* [22]. In this model, we put all the data for one week into one node as shown in Fig. 5. Next, we connect the nodes in-order to form a sequence. The sequence is then passed to a signature generation sub-module followed by the learning analytics sub-module for final determination as shown in Fig. 6.

B. Sequence Model vs Feature Vector Model

While the student sequence model uses nodes to sort and group data items temporally, a more common method uses feature vectors to represent data items. Feature vector representation in knowledge discovery and data mining constructs feature vectors for data items, in which each data item is represented by one *vector* with a fixed set of *features* or dimensions. For example, in student data, each data item could contain a vector of one student's performance in a certain course or a certain program. The features of said vector could then include the student's background information such as demographics, course information, and student's performance such as grades, assignments and activities.

Feature vector representation, or often called flat representation, makes strict assumptions for data dependencies which enables the use of conventional machine learning tools. This representation assumes that vectors are independent of each other and features are independent of other features. These independency assumptions, as well as the fixed length of the vectors (i.e. the number of features), makes the application of machine learning tools widely available.

However, having that strict assumption for data dependencies in vector representations ignores the temporal correlations in student data – something we wanted to emphasize. A typical example of such temporal correlation is the correlation between the final grade and different types of grades (e.g., class activities, lecture tests, assignments) over time for the same course. The order in which these grades occur provides important information for predicting success or risk. However, that order is discarded in feature vector representation due to its inability to represent temporal correlations.

C. Structure of the Sequence Model

For this study, our sequence model consists of 19 nodes: one node at the beginning of the semester for student background data, 17 intermediary weekly nodes which include grades and reflection responses, and one outcome node containing the overall course grade. There are four background features in the first node. The 110 grade scores and maximum of 23 reflection responses (depending on the individual student) are then spread out over the intermediary weekly nodes. We converted reflective

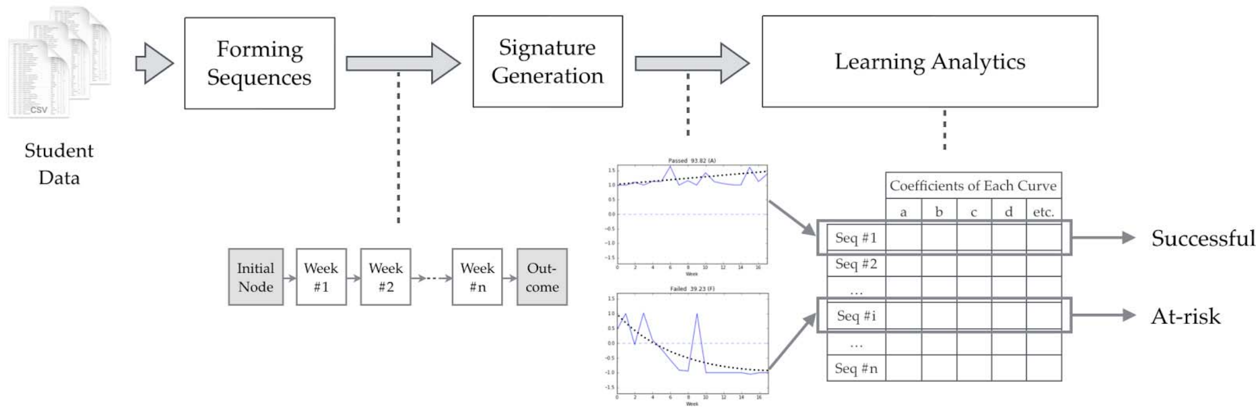

Fig. 6. The temporal analysis process using the Mahzoon et al.’ sequence data model [22].

surveys from text to numbers using the commercial linguistic sentiment analytics tool LIWC [37]. LIWC generates 93 sentiment features as numbers for every input text. Many of these features were highly correlated to each other. For this reason we chose only the 18 sentiment features with the least correlation to each other. This also improves the computational efficiency. Therefore, each reflective survey’s text was converted to a vector of 18 sentiment features.

D. Results and Discussion

One of the benefits of sequence analytics is its capability to repeat the analysis with different *salient features* to identify the predictive impact of each data category. Based on the model by Mahzoon et al., *salient features* are features that are involved in the analytics and other (or so-called) *context features* will be only used for interpretability after the analysis [22]. Our three main salient features were tests, activities, and reflections. We then experimented with these features both individually and all together to evaluate their relative predictive impact. This helps us understand the effectiveness and importance of each feature as predictors of success. For each salient feature, we ran sequence analytics to classify students at risk of obtaining a D, F, or W in the course. Fig. 7 shows two examples of individual student signatures that were generated for successful (grades of ABC) and at-risk (DFW) students. Fig. 8 shows the averages of all the student signatures in the class grouped by final grade category. Fig. 9 shows the averages grouped by successful (ABC) or at-risk (DFW). In all three figures, all data including tests, activities, and reflections were used to generate the signatures.

The classification is performed in two phases: training and validation. In the training phase of classification, we use the 10-fold cross-validation method [17] to split our data into training and validation sets. After training, the system output would be validated by repeating the validation set ten different times. The performance measures of the analytics were then averaged over the 10-fold cross-validation.

We evaluated the sequence model incrementally at multiple points in time to assess how the temporal model's accuracy changes temporally. Fig. 10 reports the model’s accuracy for the three salient features (tests, activities, and reflections) and all of them used together. These features were plotted over time to show how the accuracy improved as more data was included. In this figure, the horizontal axis shows the number of weekly nodes included in the model, and the vertical axis shows the accuracy of the model as a percentage. For example, from Fig. 10 we can conclude that even if we only use one week of the data (e.g., tests, activities or reflections) we are able to accurately classify the risk status of 70% of students. This accuracy increases as we include more nodes (i.e., more weeks into the semester) in the sequence model. However, the trend of increasing accuracy is not the same for different salient

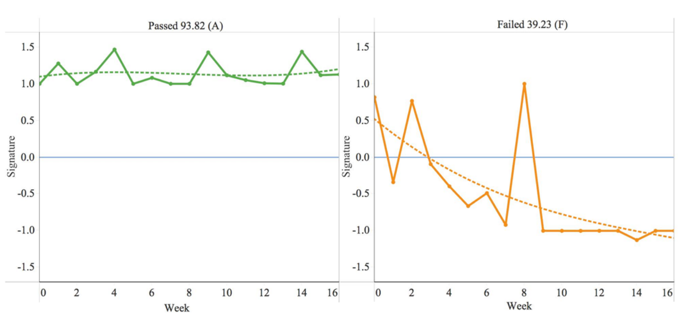

Fig. 7. An example of two different students’ signatures: the successful student (left graph) and a student failing the course (right graph). The distance to the decision boundary (dashed lines) classify students as successful (ABC) or failing (DFW). The magnitude of the distance (y-axis) represents the confidence of the classifier as weeks pass by in the semester.

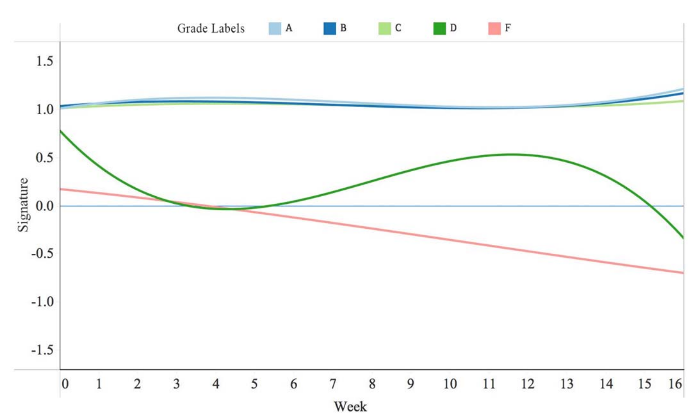

Fig. 8. Averages of signatures for all students per final grade category.

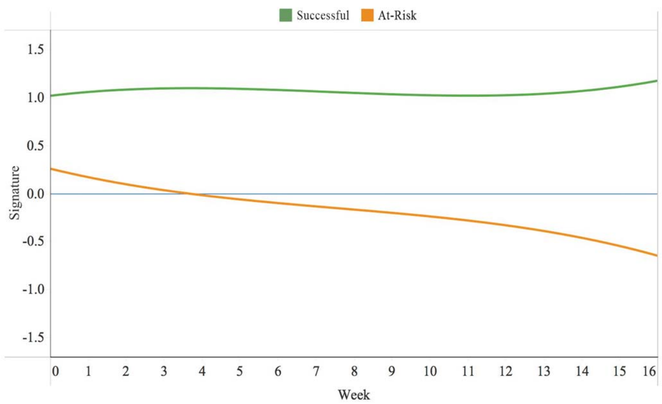

Fig. 9. Averages of signatures for all students grouped per successful (ABC grades) or at-risk category (DFW).

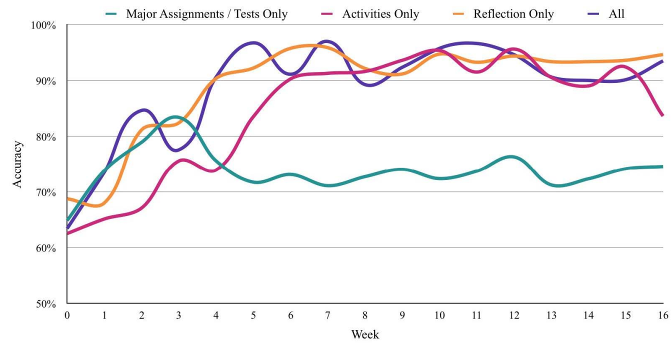

Fig. 10. Graph displaying prediction accuracies for the three salient features (tests, activities, and reflections) over time (week of the semester).

features. For instance, having tests as the only salient feature improves accuracy but only up to the four week point in the semester. On the other hand, having activities as the only salient feature produces models with higher accuracies in comparison with tests after five weeks of activities. Reflections as the salient feature performs even better than activities or tests and can predict at-risk students with 90% accuracy even after having only five weeks of reflections. In all cases, the additional benefit for including more information diminishes at the halfway point of the course. So, it is important to try to maximize earlier prediction rather than overall accuracy after the middle of the semester. At this early point, there are still opportunities to intervene on the behalf of the student. It is worth noting that the closest individual salient feature compared to including all features together in Fig. 10 is a reflections-only plot.

Based on our results, we observed that including reflections as a feature improve the accuracy of our risk classification model. This provides evidence to support both RQ1 and RQ2, that including student reflection improves accuracy and time-to-classify. This also provides additional motivation for instructors to include reflection in their classes as it is both an effective learning tool and improves our risk classification learning model.

Our findings are encouraging for integrating reflections into the curriculum. Previous research has investigated reflections as a tool for learning and has cited many different potential benefits, such as the development of metacognitive skills [39]. What we have shown in this work is that in addition to previously explored benefits affecting students, there are also benefits for instructors and administrators, such as having the ability to predict students who may be at-risk early on. With that knowledge, interventions can then be made in attempts to aid the at-risk students. Furthermore, it's even more important that these predictions can be accurately made early on, rather than later, so that it isn't too late in the course for the student to make improvements when those interventions are made.

Although our results suggest that reflections were predictive of student success on their own, they were most effective when used with traditional features such as tests, activities, and assignments. And while most features performed well after the first six weeks of the course, reflections served as the earliest predictors of success for students. Hence, for RQ3, it suggests that infusing student reflective practices between activities and throughout the course is effective as an early predictor of success. Reflection in CS has the ability to help students think more deeply about the course material and make broader connections to other courses and aspects of computing. Our work has shown that in addition to these benefits, there are also administrative benefits that help instructors and teaching staff to identify at-risk students sooner and more accurately. For instructors who are hesitant to integrate reflection into their courses, our results provide another compelling reason to integrate reflections into engineering and CS courses.

VI. Conclusions and Future Work

In this paper, a CS1 course model is proposed based on the adoption of Kolb's learning theory and the integration of iterative reflective practices in active learning settings. We applied a sequence analytics learning model to classify students based on their risk of obtaining a D, F, or W and evaluated the effectiveness of including reflections as a feature for this learning model. In doing so, we observed that reflection is an effective feature to include in increasing the accuracy of the model and decreasing the time-to-detect. These results are compelling for instructors who are considering including reflection and for learning analytics researchers who are considering features to use with their models.

Our course model integrates reflection at multiple points of the semester to encourage students to reflect on multiple aspects of their learning. This appears to be effective based on our results but future work is necessary to determine if our results generalize to other course models, different types of reflections, and different student populations. Replicating our results in these different contexts will help further validate the inclusion of reflections as a feature, our sequence-based learning model, and our course model. The optimal design of reflections is also still an open research question and this needs to be further evaluated in the context of our results. We hope that through this future work we might be able to further increase the accuracy and decrease the time-to-detect of our sequence-based learning model.

References

[1] Agudo-Peregrina, Ángel F. et al. (2014). Can We Predict Success from Log Data in VLEs? Classification of Interactions for Learning Analytics and Their Relation with Performance in VLE-Supported F2F and Online Learning. Computers in Human Behavior 31 (2014): 542. Print.

[2] Bannert, M., Reimann, P., and Sonnenberg, C. (2014). Process mining techniques for analyzing patterns and strategies in students' self-regulated learning. Metacognition and Learning, 9(2):161–185, 2014.

[3] Bonwell, C., Eison, J. (1991). Active Learning: Creating Excitement in the Classroom. Information Analyses - ERIC Clearinghouse Products (071).

[4] Boud, D.J., Churches, A.E., and Smith, E.M. (1986). Student self-assessment in an engineering design course: an evaluation, International Journal of Applied Engineering Education, 2, 2, 83-90.

[5] Boud, D., Keogh, R., and Walker, D., (1985). Promoting reflection in learning: A model. Reflection: Turning experience into learning: 18-40.

[6] Bourner, T. (2003). Assessing Reflective Learning. Education + Training. 45.5 (2003): 267-272. Print.

[7] Cherenkova, Y., Zingaro, D., and Petersen, A., (2014). Identifying challenging CS1 concepts in a large problem dataset. In *Proceedings of the 45th ACM technical symposium on Computer science education* (SIGCSE '14). ACM, New York, NY, USA, 695-700.

[8] Dochy, F. J. R. C., Mien Segers, and Dominique Sluijsmans. (1999). The use of self-, peer and co-assessment in higher education: A review. Studies in Higher education 24.3 (1999): 331-350.

[9] Dorodchi, M., Dehbozorgi, N. (2017). Addressing the Paradox of Fun and Rigor in Learning Programming. In Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '17).

[10] Duderstadt, J. J. (2008). Engineering for a Changing World: A Roadmap to the Future of Engineering Practice, Research, and Education. The Millennium Project. Ann Arbor, MI: The University of Michigan.

[11] Fink, L. D. (2003). A Self-Directed Guide to Designing Courses for Significant Learning. (Notes based on Fink, L. Dee. 2003. Creating significant learning experiences: An integrated approach to designing college courses. San Francisco: Jossey-Bass).

[12] Franke, R. H., and Kaul, J. D. (1978). The Hawthorne experiments: First statistical interpretation. American sociological review (1978): 623-643.

[13] Guzdial, M. (2015). Bringing evidence-based education to CS. Commun. ACM 58, 6 (May 2015), 10-11.

[14] Harb, J. N., and Terry, R. E. (1992). A Look at Performance Tools through the Use of the Kolb Leaning Cycle, 1992 ASEE Annuul Conference Proceedings

[15] Howard, R. A., Carver, C. A., and Lane, W. D., (1996). Felder's learning styles, Bloom's taxonomy, and the Kolb learning cycle: tying it all together in the CS2 course. *ACM SIGCSE Bulletin*. Vol. 28. No. 1. ACM.

[16] Kapur, M., (2011). Temporality matters: Advancing a method for analyzing problem- solving processes in a computer-supported collaborative environment. International Journal of Computer-Supported Collaborative Learning, 6(1):39–56, 2011.

[17] Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In IJCAI (Vol. 14, No. 2, pp. 1137-1145).

[18] Kolb, David A. (1976). Management and the learning process. California management review 18.3 (1976): 21-31.

[19] Kolb, David A. (2014). Experiential learning: Experience as the source of learning and development. FT press, 2014.

[20] Kulkarni, Chinmay, et al. (2015). Peer and self assessment in massive online classes. Design thinking research. Springer, Cham, 2015. 131-168.

[21] Lishinski, A., Yadav, A., and Enbody, R. (2017). Students' Emotional Reactions to Programming Projects in Introduction to Programming: Measurement Approach and Influence on Learning Outcomes. In Proceedings of the 2017 ACM Conference on International Computing Education Research (ICER '17).

[22] Mahzoon, M. J., Maher, M. L., Eltayeby, O., Dou, W., & Grace, K. (2018). A Sequence Data Model for Analyzing Temporal Patterns of Student Data. Journal of Learning Analytics, 5(1), 55-74.

[23] McMillan, J. H., and Hearn, J. (2008). Student Self-Assessment: The Key to Stronger Student Motivation and Higher Achievement. *Educational Horizons*, vol. 87, no. 1, 2008, pp. 40–49. *JSTOR*, JSTOR, www.jstor.org/stable/42923742.

[24] Metcalfe, J., & Shimamura, A. P. (1994). Metacognition: Knowing about Knowing. Cambridge, MA: MIT Press.

[25] Mohamad, S. K., and Tasir, Z. (2013). Educational data mining: A review. Procedia - Social and Behavioral Sciences, 97:320 – 324, 2013. The 9th International Conference on Cognitive Science.

[26] Molenaar, I., (2014). Advances in temporal analysis in learning and instruction. Frontline Learning Research, 2(4):15–24, 2014.

[27] Murali Mani and Quamrul Mazumder. 2013. Incorporating metacognition into learning. In Proceeding of the 44th ACM technical symposium on Computer science education (SIGCSE '13). ACM, New York, NY, 53-58.

[28] Pellegrino, J. W., Chudowsky, N., and Glaser, R. (editors). (2001). Knowing what students know: The science and design of educational assessment. Washington, DC: National Academy Press.

[29] Penã-Ayala, A. (2014). Educational data mining: A survey and a data mining-based analysis of recent works. Expert Systems with Applications, 41(4, Part 1):1432 – 1462, 2014.

[30] Piaget, J., and Cook, M. (1952) The origins of intelligence in children. Vol. 8. No. 5. New York: International Universities Press, 1952.

[31] Reimann, P., (2009). Time is precious: Variable and event-centred approaches to process analysis in CSCL research. International Journal of Computer-Supported Collaborative Learning, 4(3):239–257, 2009.

[32] Sanders, K., et al. (2017). Folk Pedagogy: Nobody Doesn't Like Active Learning. In Proceedings of the 2017 ACM Conference on International Computing Education Research (ICER '17). ACM, New York, 145-154.

[33] Schatzberg, L. (2002). Applying Bloom's and Kolb's theories to teaching systems analysis and design. In *The Proceedings of ISECON* (Vol. 19).

[34] Sepp, Lauren A., et al. (2015). On an upward trend: Reflection in engineering education." Proceedings of 2015 American Society of Engineering Education Annual Conference. 2015.

[35] Stone, J. (2012). Using Reflective Blogs for Pedagogical Feedback in CS1. ACM, 2 Penn Plaza, Suite 701, New York, NY 10121-0701, USA, 2012.

[36] Sukhija, K., Jindal, M., & Aggarwal, N. (2015). The recent state of educational data mining: A survey and future visions. In MOOCs, Innovation and Technology in Education (MITE), 2015 IEEE 3rd International Conference on (pp. 354-359). IEEE.

[37] Tausczik, Y.R., & Pennebaker, J.W. (2010). The psychological meaning of words: LIWC and computerized text analysis methods. Journal of Language and Social Psychology, 29, 24-54.

[38] Topping, K. (1998).Peer assessment between students in colleges and universities." Review of educational Research 68.3 (1998): 249-276.

[39] Turns, J. A., Sattler, B., Yasuhara, K., Borgford-Parnell, J. L., and Atman, C. J., (2014). Integrating Reflection on Experience into Engineering Education. Proc. of the ASEE Annual Conf. & Exposition.

[40] Wadsworth, B. J. (1996). Piaget's theory of cognitive and affective development: Foundations of constructivism. Longman Publishing, 1996.

[41] Whitehurst, G.. (2002). Archived: What is EBE? Evidence-Based Education (EBE) Presentation. U.S. Department of Education, Oct. 2002, Retrieved on 8/20/2017 from https://www2.ed.gov/nclb/methods/whatworks/eb/edlite-slide003.html.

[42] Wolff, A., Zdrahal, Z., Nikolov, A., & Pantucek, M. (2013). Improving retention: predicting at-risk students by analysing clicking behaviour in a virtual learning environment. In *Proceedings of the third international conference on learning analytics and knowledge* (pp. 145-149). ACM.

[43] Xinogalos, S. (2016). Designing and deploying programming courses: Strategies, tools, difficulties and pedagogy. Educ Inf Technol. 21: 559.

[44] You, Ji Won. (2016). Identifying Significant Indicators Using LMS Data to Predict Course Achievement in Online Learning. The Internet and Higher Education 29 (2016): 23–30.