

The Relationship Between Computer Science Instructional Practices and Retention-A Multi Level Study

¹ Z. Shobharani ² Y. Bhaskar Reddy ³Dr. B. Sudarshan

^{1&2} Assistant professor Department of computer science engineering,
³ Associate Professor, Dept., of Mechanical Engineering
K.S.R.M. College of engineering, kadapa-AndhraPradesh-India-516003

Abstract:- Many computer science (CS) departments want to increase student retention in their courses. Understanding the factors that influence the probability of students continuing to enroll in CS courses is a critical step in increasing retention. Prior research on CS retention has mainly focused on variables like prior programming experience and students' personality traits, all of which are outside the control of undergraduate instructors. This research looks at factors that are under the influence of teachers, specifically instructional practices that have a direct effect on students' classroom experiences. Over the course of four semesters, participants were recruited from 25 parts of 14 different courses. Although adjusting for students' mastery of CS concepts and status as a CS major, a multi-level model was used to assess the effects of individual and class-average perceptions of cooperative learning and instructor directedness on the likelihood of subsequent enrollment in a CS course. The average rating of cooperative learning within a course segment was negatively correlated with retention, according to the findings. Students' individual impressions of instructional activities were not associated with retention. Greater mastery of CS concepts and considering or having declared a CS major were linked to a higher likelihood of taking potential CS courses, which is consistent with previous study. The findings' implications are explored.

Keywords:- Computer science education; Retention; Multi-level models; Co-operative learning; Student-centered classrooms.

1. INTRODUCTION:

In recent years, there has been a surge of interest and energy in increasing computer science (CS) education at all levels in the United States. There is the agreement that awareness of CS principles, computational thinking skills, and experience with coding will play an important role in future workforce readiness, from the CS for All consortium in the United States to the increase in European countries that have added coding to the elementary and secondary school curriculum [6]. It is critical for tertiary CS programmers to recruit and retain students in order for the United States to fulfill potential workforce demands for CS professionals [15]. At the undergraduate level, however, significant proportions of students in STEM disciplines have traditionally failed to complete the degree they started [26, 30], and CS is no exception [9]. A history of low retention and degree completion rates in computer science [13, 14] has prompted research into the factors that influence retention, as knowing these factors can help educators develop strategies to recruit and maintain

students in CS programmers. Prior exposure to CS and programming [13], personality traits [1], motivation for a specific course [23], the learning atmosphere [9], and achievement in introductory CS courses [13] are all variables that researchers have discovered. That indicates whether students will continue to take CS classes. Students' experiences in classes within a discipline can also influence whether or not they choose to take classes [2, 5, and 31].

We are not aware of any studies that have looked into the effect of CS students' impressions of instructional activities on their retention in CS classes. The aim of this study is to close this gap in the literature. As teachers, programmers, and universities work to increase retention rates, it's critical to understand the factors that affect retention that are beyond educators' influence. Many areas of education are under the full or partial control of the government. Instructional practices are a potentially promising environment for approaches that can affect retention because the instructor has near complete control.

The aim of this study was to see how (a) students' perceptions of an information-transmission orientation to teaching (later called instructor directedness, TD) and (b) students' perceptions of peer cooperation and engagement within a course section affect retention in CS courses (later called cooperative learning, CL). The study's main research question was whether students' expectations of instructional activities influence their probability of enrolling in at least one CS course the following semester. We'll go through some of the research that has been done on this topic in the following section. The study is then described, followed by the study's conclusions, a review of the findings and their consequences, and finally, concluding remarks.

This study adds to the CS education literature by answering the key research question and illustrating the use of a multi-level logistic regression model in a CS education environment. MLMs are used in education research to account for data "clustering," which occurs normally in samples of students who are clustered within a collection of CS; Students were recruited from 25 separate course sections over the course of four semesters for this research. Individuals within a cluster appear to be more similar to one another than individuals from different clusters because of the mutual impact of their cluster's climate. Failing to account for this form of clustering within a dataset biases statistical estimates because individuals within a cluster tend to be more similar to one another than

individuals from different clusters (in this case, being in the same course section). As CS education study expands and studies draw broader samples from a variety of courses and organizations. Students had more justifications for cheating when they (a) felt a personal attachment to their teacher, (b) showed less involvement during classes, (c) felt less collegiality with classmates, (d) were less pleased with the course, (e) thought classes were less coordinated, and (f) felt less autonomy and power over their work in the course. Undergraduate chemistry students' perceptions of the relative importance of learning and evaluation outcomes (e.g., grades), cognitive engagingness of lectures, and harshness of grading practices were linked to the types of achievement goals they set for the course, which influenced their achievement and motivation for the course [3]. Students who thought their instructor was more concerned with grades than with learning were less likely to set mastery goals and were more likely to set goals linked to their final score and others' perceptions of their competence (performance goals). Students who thought grading methods were harsh were less likely to set mastery goals and focused more on not failing (performance-avoidance, as opposed to goals to work toward a desirable grade, performance-approach). Finally, students who thought lectures were interesting were more likely to set mastery goals rather than avoidance goals. Mastery goals were positively related to both outcomes, performance avoidance goals were negatively related to both, and performance solution goals were positively related to grades. Finally, [32] looked into the links between self-regulated thought, motivation, and actions and marketing students' perceptions of the classroom "environment." The classroom environment predicted students' achievement target orientation, perceived competence, and perceived autonomy in the course. Furthermore, students' views of the course environment in terms of grades and performance assessment are predictive of their use of preparation techniques while preparing for the course.

2. RELATED RESEARCHES:

2.1 *The Classroom's Perceptions*

Students' views of the classroom have an impact on their actions in that class. Students' perceptions of their course can differ significantly from teachers' beliefs about students' perceptions of their course. For example, any instructor who has been surprised by end-of-term student evaluations will attest to this difference. Since teachers are not always able to determine the environment of their classroom or how their teaching is received critically, it is critical to understand the position of students' perceptions. Researchers from a number of fields have looked into the effects of the perceived classroom climate on a variety of outcomes. [25], for example, looked at the relationship between college students' perceptions of the classroom, cheating, and justifications for cheating. Students' classroom experiences were linked to cheating in a variety of disciplines. Students who thought the course was more structured and were happier with their experiences in it were less likely to cheat. Perceptions in the classroom were often linked to justifications for cheating, regardless of

whether or not the student had cheated (or admitted to cheating).

2.2 *Retention-focused interventions*

Many initiatives aimed at increasing retention in science, technology, engineering, and mathematics (STEM) disciplines have been introduced, analyzed, and published in peer-reviewed journals. Wilson et al. [31], for example, identified a program at Louisiana State University's Howard Hughes Medical Institute (LSU-HHMI). Via a multi-tiered mentoring model, the LSU-HHMI Professors Program combined mentorship, undergraduate research experiences, and supplemental academic and professional growth opportunities. Students in the mentoring program had higher six-year graduation rates than STEM students at LSU who were not in the program and STEM students around the country who participated in a STEM program the same year.

Dagley et al. [5] identified a STEM learning culture that improved undergraduate retention. Residential (an on-campus learning community), social (STEM-oriented social events), and curricular (undergraduate research activities and cohort-based math courses) components are all part of the two-year EXCEL curriculum at the University of Central Florida. An examination of student retention over a number of years revealed. the EXCEL students were more likely to remain enrolled in their STEM major one year after matriculation and to graduate than comparable STEM students who did not participate in the EXCEL program during the same time span.

While programs like those mentioned in [31] and [5] have been shown to be successful, they are complex, resource-intensive programs that can only be introduced if key staff at multiple levels of an organization agrees to commit the necessary resources. Smaller interventions that can be undertaken by one or a few dedicated individuals are a more realistic choice when budgets are tight or being slashed, or when other obstacles to large-scale reform exist. Academic units are being asked to do more for a growing number of students with limited financial resources in today's climate. As a result, efforts to improve retention that are less resource-intensive than robust initiatives like the ones mentioned above are becoming increasingly important. Changes at the program or department level, as well as instructional choices at the classroom level, are two options for increasing retention while putting a low demand on resources.

The retention of CS majors has been shown to be improved by a few program-level improvements and classroom interventions. Ott et al. [21], for example, recorded a program-level shift that affected student retention. Two versions of introductory CS were offered at the research institution: one for students with prior programming experience and another for students with no prior programming experience. Students could opt-in to the course for students with programming experience for many years, but only a small percentage of students with programming experience did so. More students enrolled in the course that was suitable for their level of previous experience when they were asked to take a placement test, and the improvement in placement procedures improved

the likelihood that students would continue taking CS courses. Pair programming was used during an introductory course's laboratory exercises in a class-room intervention recorded by Carver et al. [2]. Pair programming is a joint approach to programming tasks in which two students take turns acting as the programmer or "driver" and the debugger or "navigator" [28]. Students who participated in pair programming activities in introductory programming courses were more likely to continue in their computing major than students who did not participate in pair programming activities in the same courses [2]. In each of these cases, a heavily CS-related intervention was effective in improving retention among students majoring in computing. Our interest, on the other hand, is in deciding whether some aspects of instructional practices influence whether students enroll in additional CS courses.

In comparison to the large-scale projects outlined by [5, 31], classroom-level improvements that can affect student retention, such as choices related to instructional strategies, are relatively low-cost and easy to introduce. For example, the model proposed by Graham et al. [10] defined active learning instruction as a factor in students' persistence in STEM programs. Active-learning teaching, which may involve cooperative learning (CL), is a low-cost way to potentially improve student performance, all students, majors and non-majors alike, would benefit from improved learning opportunities. It's likely that many elements of teaching are linked to students' willingness to continue taking CS courses. Understanding how instructional activities are linked to retention will help teachers incorporate classroom-level strategies to improve CS retention.

3. METHODOLOGIES

The university's IRB approved the report, and students volunteered to participate in data collection. The data for this thesis came from a broader NSF-funded study of computational thought in undergraduate computer science. This section only covers the data collection and materials used in the current research.

3.1 Participants and Methodology

Undergraduate students enrolled in computer science courses at a big, public Midwestern university (N = 607; 502 males, 105 females) took part in the study. Over four semesters, students from five 100-level courses (n = 420), one 200-level course (n = 64), three 300-level courses (n = 82), and five 400-level courses (n = 41) were recruited from five 100-level courses (n = 420), one 200-level course (n = 64), three 300-level courses (n = 82), and five 400-level courses (n = 41). The total number of distinct sections (i.e., clusters in the analysis) was 25.

The first week of the semester, around the eighth week (depending on the timing of each course's events), and the week before final exams were all used to collect survey-based results. Students' motivation, self-regulation, course-related affect, learning habits, and impressions of classroom teaching, as well as demographic data, were all measured in the surveys. A test of core CS concepts was

also included in the end-of-semester survey (described in Section 3.2).

Participants had the choice of allowing the researchers access to their potential course enrollment when they agreed to participate in the study. After the open drop/add cycle for that semester had ended, enrollment data was collected from the university. Because of the various reasons students could not enroll in classes during the semester following their participation in the study, students who did not enroll in any classes during the semester following their participation in the study were not included in the review (e.g., graduation, transferring, drop-ping out of school). As a result, the distinction is between students who enrolled in a CS course and students who were enrolled at the university during that semester but did not enroll in a CS course. Enrollment data was dichotomized to show whether students enrolled in a CS course the following semester (enrollment = 1) or did not (enrollment = 0). Students who re-enrolled in the course from which they were recruited were not counted; students who retook a course had to be enrolled in at least one other course to be counted as enrolling in CS that semester. 53.4 percent of the participants (n = 356) enrolled in a computer science course the next semester.

3.2 Predictor Variables

The Student Perceptions of Classroom Knowledge Building (SPOCK) scale was used to assess student perceptions of instructional activities. The SPOCK is a course-specific instrument that assesses students' self-control and use of learning techniques, as well as their question-asking habits and impressions of the classroom environment, like CL and TD. Only the elements related to the classroom environment (Cooperative Learning and Teacher Directedness subscales) were included in the study, which was based on a condensed version of the SPOCK that has been used in other studies [7, 27]. There are 27 elements in the condensed edition that make up the same subscales as the complete version. The SPOCK includes items with a 5-point answer scale ranging from "almost never" to "almost always." To help respondents understand the labels, each answer category has a brief explanation (e.g., Sometimes, occurred frequently: occurred around 34% of the time).

Three things were included in the Cooperative Learning subscale ("In this class, my classmates and I actively collaborated to help each other understand the material.") "My classmates and I actively collaborated to complete assignments in this class." "I received positive remarks about my work from other students in this class while I was doing my work.") The alpha coefficient for this sample was .82. Three elements were included in the Teacher Directedness subscale ("In this lesson, the instructor concentrated on getting us to learn the correct answers to questions.") "The teacher instructed us what the most valuable knowledge was in this class." "The teacher gave us detailed guidance about what we were supposed to do in this class.") The alpha coefficient for this sample was .78.

The relationship between students' mastery of CS content and retention was controlled for using scores on the

Nebraska Assessment of Computing Knowledge (NACK). The NACK is a 13-item multiple-choice test that covers basic computer science concepts. It acted as a standardized measure of CS mastery through courses and instructors. Previous studies have used the test, and details on its growth can be found in [20, 27]. This sample had a coefficient alpha of .78.

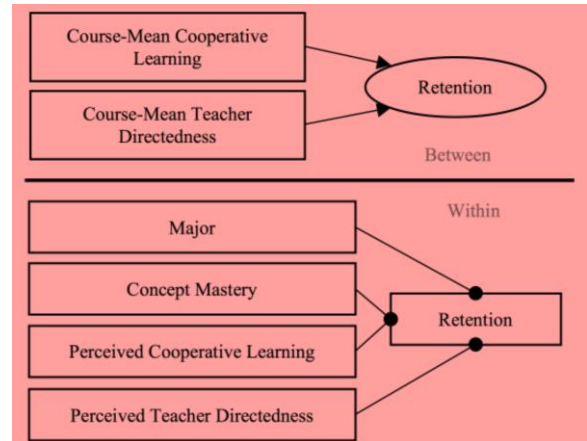
The relationship between one's major and retention was controlled for using the participants' self-report of their status as a declared CS major. "Are you thinking about majoring or minoring in Computer Science/Computer Engineering?" "Yes," "No," and "I am already majoring/minoring in Computer Science/Computer Engineering" were the answer choices. We included major status as a control in our model because declared or expected major/minor is one of the greatest influences on the classes one takes. The three choices were dummy coded into two binary variables: one for "considering" status (considering = 1, else = 0), and one for "majoring" status (major = 1, else = 0), leaving the intercept as those who were not considering or majoring in computer science or computer engineering.

3.3 Multi-level Models (MLMs)

Samples are often taken from naturally clustered systems in education study, such as students nested within courses that are nested within schools. Clustering introduces bias into statistical calculations because individuals in the same cluster are more likely to be identical (in terms of measured and unmeasured variables) than those in different clusters. Multi-level models [8,12] can be used with a variety of single-level statistical methods, such as regression, random-coefficient models, and structural equation modeling, to account for multiple levels of clustering within results (SEM).

When creating an MLM, relationships between variables are defined at the within- or micro-level (also known as Level 1), between- or macro-level(s) (also known as Level 2, Level 3, etc.), and probably between levels. Students were nested within course sections in this review, so Level 1 was the student level and Level 2 was the section level (there were not enough unique courses to model a third level for course). Level 1 predictor variables—perceptions of instructional activities, major status, and achievement—and Level 1 outcomes—enrollment in a CS course the following semester—are defined in the Level 1 portion of our model. Our model's Level 2 predictors—the section mean of perceived instructional practices—and Level 2 outcomes—the section mean for enrolling in a CS course the following semester—define relationships between the two. Since enrollment was coded as a binary variable, a logistic regression model was used. A visual representation of the model is shown in Figure 1.

The intra-class correlation is a statistic that MLMs are interested in (ICC).



The ICC measures how much clustering in the model accounts for uncertainty in the outcome (in this case, retention) (here, course sections). The ICC will be similar to 1 if there are significant variations in the outcome variable between groups (course sections) but only minor differences within each group. On the other hand, if there are significant variations between groups and group means are virtually equal, the ICC would be close to zero. In this analysis, a high ICC indicates that the majority of students in some course sections took subsequent CS courses while the majority of students in other course sections did not, and a low ICC indicates that the rate of retention was comparable across all course sections.

R², regression coefficients, and odds ratios are other statistics of interest for the model used in this analysis. The interpretation of R² in logistic regression is close to that of regression with continuous outcomes. The adjustment in the natural log of the probability that the outcome will occur (here, enrollment = 1) that is associated with every 1-unit increase in the predictor variable is estimated by logistic regression coefficients. The odds ratio, on the other hand, shows the shift in the odds of an outcome occurring as a result of a 1-unit rise in the predictor variable and is easier to comprehend. For example, in this study, a 2.0 odds ratio for majoring or not majoring in CS meant that students already majoring in CS ($x = 1$) were twice as likely to take a CS course as students not majoring in CS ($x = 0$).

Finally, the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC) were used for model comparison (BIC). To compare the fit of similar models, AIC and BIC can be used. They consider model fit (in terms of log likelihood) and model complexity (in terms of estimated parameters), with BIC penalizing additional parameters more severely. Smaller values mean a better fitting model for both statistics, but there are no conventions or cutoffs for interpreting them as there are for any other fit statistics since they are strongly affected by the existence of the tested model.

The research for this study was done in Mplus V. 8.1 [18], with the analysis command "TYPE = TWOLEVEL," which indicated a two-level clustering structure with randomly varying intercepts for each cluster (i.e., a separate cluster mean was estimated for each cluster). The cluster variable specified the specific course section from which participants were drawn. A random intercept for

course section is a way to account for uniqueness associated with a given cluster, such as the instructor, the course stage, and the course section's shared atmosphere. Level 1 included student ratings of CL and TD that were based within-cluster so that approximate parameters reflected a student's view of the world in relation to his or her classmates. This form of centering also allows you to distinguish between individual-level and class-level effects [16]. Level 2 included cluster means as CL and TD measures, capturing the relationship between average CL and TD ratings and the proportion of students in a cluster who enrolled in a CS course the following semester. The inclusion of centered-within-cluster parameters and group mean parameters enabled us to model the effect of the individual's perception on retention as well as the effect of the actual instructional setting, assuming that group average ratings of instructional practices approximate actual instructional practices

When using aggregations of individual ratings as indicators of a classroom-level variable, the classroom-level variable's reliability must be checked [16]. This is accomplished by estimating the ICC for the variable and then adjusting the ICC using the Spearman-Brown formula based on the average number of units (in these case, students) per cluster (Eq. 2 in [16]). If the cluster size grows, so does the estimation of aggregate variable reliability, which is similar to the predicted increase in test reliability associated with raising the number of items on the test.

4. CONCLUDING REMARKS

4.1 Baseline Model

To begin, we ran a baseline model to see how well participants' segment predicted their enrollment. The empty model had an ICC of 0.734, meaning that the course portion in which participants were enrolled accounted for the majority of enrollment variability.

Table 1: Model Statistics

Model	AIC	BIC	ICC
Empty	504.77	513.77	0.734
Full	363.33	403.13	0.476

Table 2: Logistic Regression Statistics for the Full

Model					
B	S.E.	Wald	p	Odds ratio	
Level 1					
CL*	0.17	0.15	1.14	.256	1.18
TD*	-0.18	0.20	-0.89	.374	0.84
CS	0.14	0.06	2.47	.013	1.15
Knowledge Test					
Considering Major	3.10	0.35	8.80	<.001	22.11
Already a Major	3.81	0.66	5.74	<.001	44.98
Level 2					
CL**	-1.33	0.66	-2.01	.044	--
TD**	0.12	0.90	0.13	.893	--

Note. B = regression coefficients, S.E. = standard error, CL = cooperative learning, TD = teacher directedness, *centered-in-cluster, **cluster mean.

Table 1 shows the ICCs as well as fit statistics for both versions. Students in advanced courses were undoubtedly CS majors who would take additional CS courses, and some of the introductory sections at this university were mainly taken by non-CS majors, so the high ICC was unsurprising. We expected the ICC to be lower once the predictors were applied since our hypothesized model included Level 2 predictors (i.e., predictors related to characteristics of the different course sections).

4.2 Reliability of Aggregate Variables

The overall classroom awareness variables' reliability was then assessed. Following the steps outlined in Section 3.3, this was accomplished. ICC = 0.332 in Cooperative Learning and 0.203 in Teacher Directedness. Each cluster had an average of 26.68 students in it. As a result, CL = 0.930 and TD = 0.872 were calculated as reliability estimates for the class means.

4.3 Hypothesized Model

The "full" hypothesized model was then checked, which included all of the predictor variables. The complete model approximated the data better than the empty model, according to model fit statistics. The ICC for the full model was lower than the empty model, ICC = 0.476, as predicted, suggesting that the collection of predictors only partially explained the differences between clusters. Table 2 lists the predictors and their statistics.

The control variables were important, as predicted. Majors were 45 times more likely to enroll in a CS course, and those considering a CS major were 22 times more likely than non-majors. Higher mastery of course material was linked to a small increase in the probability of taking further CS courses.

CL and TD expectations at Level 1 were not relevant.

A lower probability of taking additional CS courses was correlated with a higher cluster mean rating of CL.

5. DISCUSSION AND IMPLICATIONS

This paper shows how an MLM can be used in CS education research. MLMs are needed when samples are drawn from multiple courses or institutions due to the normal nesting that occurs in educational settings. If the area of computer science education expands and broader studies are conducted, researchers may need to become acquainted with and incorporate MLMs.

Individual perceptions of instructional activities were not linked to retention, which was a major finding in this study. A student who perceives the classroom to be more peer-directed or teacher-directed is neither more nor less likely to take additional CS courses than a student who perceives the same classroom setting to be less so. The aggregate CL rating in a course segment was correlated with a lower retention rate, according to a second major finding of this study. This surprising relationship may be due to a greater focus on using CL in introductory classes, which have a higher number of non-major students who are less likely to take CS b.

Reviewing the CL course section means showed that CS1 course sections appeared to be more comparable in their overall CL ratings (Ms ranging from 2.43 to 3.56) than upper-level courses, not that CS1 courses were consistently higher in using CL (Ms from 1.57 to 3.86). At this time, it's unclear how this variation in the use of CL in upper-level courses affects students' perceptions and satisfaction with their courses, but future studies should look into this.

The perceived instructional practices indicators used in this analysis were common in nature and did not include any of the CS-specific practices that are prevalent in the literature. More relevant instructional methods, such as active learning [17], pair programming [24, 29], game-based learning [22], context-based teaching [4], and the use of multimedia [11], should be investigated in future study. Finally, although general instructional activities were not predictive of CS retention, concept mastery was, emphasizing the value of using high-quality, evidence-based teaching practices that increase the probability of students mastering CS concepts and skills.

Unsurprisingly, students' status as a major or considering a major in CS was the best indicator of their continued enrollment in CS courses. Since students' majors have such a significant impact on the courses they take, it is important to consider major when researching retention in a particular discipline. The sample in this study included students in introductory CS (CS1) courses that were required for their non-CS major, with engineering students comprising a substantial portion of students in the CS1 course sections. It also included CS1 sections for CS majors and upper-level CS courses that were overwhelmingly taken by CS majors. Even after accounting for the different sections students were in (by using a MLM), students' individual status as a major, non-major, or considering a major strongly predicted whether they would take an additional CS course. It should be noted, however, that the modes tested in this study did not include random slopes, and thus specified that the relationship between the instructional practices variables and retention is the same for all of the courses. Future research (with a larger sample of course sections) should consider the possibility of differences in this relationship by including random slope parameters.

The students in this study were enrolled in introductory computer science (CS1) courses as part of their non-CS major requirements.

In the CS1 course sections, engineering students make up a significant portion of the students. It also included CS1 parts for CS majors as well as upper-level CS courses that CS majors overwhelmingly took. Students' individual status as a major, non-major, or considering a major strongly predicted whether they would take an additional CS course, even after accounting for the various sections they were in (using an MLM). Students' individual status as a major, non-major, or considering a major strongly predicted whether they would take an additional CS course, even after accounting for the various sections they were in (using an MLM). However, it should be noted that the modes used in this analysis did not involve random slopes,

indicating that the relationship between instructional practices variables and retention is consistent across all courses. The probability of variations in this relationship should be considered in future research (with a wider sample of course sections) by using random slope parameters.

Students who scored higher on the core CS concepts exam were also more likely to continue in CS classes. However, this study cannot ascertain the causality of this relationship: it is possible that students who have a greater mastery of the course interpret that mastery as a signal that they should precede in the discipline, but it is also possible that students who have chosen to take several CS courses are more likely to master the material. Both theories are, in all probability, at least partially correct.

Given the above findings, we suggest that students' decisions to take additional courses in a discipline are affected more by their major and future career plan, which have been linked to retention in CS [19], rather than the instructional practices used in any particular course. It should be noted that there is a wealth of research suggesting that instructional practices have an effect on students' learning; this explanation does not refute the literature, but rather indicates that the relationship between instructional practices and learning is distinct from the relationship between instructional practices and retention.

6. CONCLUSIONS

As CS educators continue to be concerned about rising undergraduate retention rates, it is critical for the group to gain a deeper understanding of the factors that influence performance and retention in the field. This study looked at how four variables, including students' expectations of instructional practices, class level measures of instructional practices, students' status as a CS major, and students' mastery of course material, are linked to the probability of students continuing to take CS classes. After adjusting for students' individual plans for majoring in CS and their mastery of CS concepts, the general instructional practices used in the course section do not predict retention, according to the findings. Additional research into other CS-specific instructional practices could aid CS educators in identifying those that inspire students to continue taking CS courses.

The use of multilevel modeling is one of the study's contributions. It would be impossible to evaluate the relationships between course-level variables over a collection of courses without multi-level modeling, and person-level estimates would be inaccurate. The course-level effect of CL would not have been observed if only the person-level variables had been used in this analysis.

Future research should look into the importance of content mastery and students' majors in retention, in addition to looking into the influence of other instructional activities. With content mastery, a closer look at the aspects of core CS content that predict retention and performance in later classes may reveal how course topics can be timed, revised, and combined with other retention-boosting strategies. Considering the likelihood of different impacts of instructional activities on CS majors and non-CS majors in

terms of retention could offer insights into which practices are better adapted for which groups of students. If majors and non-majors have different relationships between instructional practices and retention, this suggests that some practices are better suited for CS1 courses and others for later courses. Overall, there is still a lot to learn about the relationship between instructional activities and CS retention.

REFERENCES

- [1] S. Beyer. 2014. Why are women underrepresented in Computer Science? Gender differences in stereotypes, self-efficacy, values, and interests and predictors of future CS course-taking and grades. *Computer Science Education*, 24(2-3), 153-192.
- [2] J. C. Carver, L. Henderson, L. He, J. Hodges, and D. Reese. 2007, July. Increased retention of early computer science and software engineering students using pair programming. In *20th Conference on Software Engineering Education & Training (CSEET'07)* (pp. 115-122). IEEE.
- [3] M. A. Church, A. J. Eliot, and S. L. Gabel. 2001. Perceptions of classroom environment, achievement goals, and achievement outcomes. *Journal of educational psychology*, 93(1), 43
- [4] S. Cooper, and S. Cunningham. 2010. Teaching Computer Science in Context, *ACM Inroads*, 1(1), 5-8.
- [5] M. Dagley, M. Georgiopoulos, A. Reece, and C. Young. 2016. Increasing retention and graduation rates through a STEM learning community. *Journal of College Student Retention: Research, Theory & Practice*, 18(2), 167-182.
- [6] Erective. 2015. Infographic: Coding at school - How do EU countries compare? Retrieved from <https://www.euractiv.com/section/digital/infographic/infographic-coding-at-school-how-do-eu-countries-compare/>
- [7] A. E. Flanigan, M. S. Peteranetz, D. F. Shell, and L.-K. Soh. 2017. Implicit intelligence beliefs of computer science students: Exploring change across the semester. *Contemporary Educational Psychology*, 48, 179-196.
- [8] A. Gelman and J. Hill. 2006. *Data analysis using regression and multilevel/hierarchical models*. Cambridge University Press.
- [9] M. N. Giannakos, I. O. Pappas, L. Jaccheri, and D. G. Sampson. 2016. Understanding student retention in computer science education: The role of environment, gains, barriers and usefulness. *Education and Information Technologies*, 1-18.
- [10] M. J. Graham, J. Frederick, A. Byers-Winston, A. B. Hunter, and J. Handel man. 2013. Increasing persistence of college students in STEM. *Science*, 341(6153), 1455-1456.
- [11] M. Guzdial, and E. Soloway. 2002. Teaching the Nintendo Generation to Program, *Communications of the ACM*, 45(4):17-21.
- [12] R. H. Heck, and S. L. Thomas. 2015. An introduction to multilevel modeling techniques: MLM and SEM approaches using Mplus. Routledge
- [13] S. Katz, D. Allbritton, J. Aronis, C. Wilson, and M. L. Soffa. 2006. Gender, achievement, and persistence in an undergraduate computer science program, *The DATA Base for Advances in Information Systems*, 37(4), 42-57.
- [14] S. Katz, J. Aronis, D. Allbritton, C. Wilson, and M. L. Soffa. 2003. Gender and race in predicting achievement in computer science, *IEEE Technology and Society Magazine*, 22(3), 20-27.
- [15] D. Langdon, G. McKittrick, D. Beede, B. Khan, and M. Doms. 2011. *STEM: Good jobs now and for the future (Issue Brief #03-11)*. Washington, DC: U.S. Department of Commerce, Economics and Statistics Administration.
- [16] O. Lüdtke, A. Robitzsch, U. Trautwein, and M. Kunter. 2009. Assessing the impact of learning environments: How to use student ratings of classroom or school characteristics in multilevel modeling. *Contemporary Educational Psychology*, 34(2), 120-131.
- [17] J. J. McConnell. 1996. Active learning and its use in computer science. *ACM SIGCSE Bulletin*, 28(SI), 52-54.
- [18] L. K. Muthén and B. O. Muthén. 1998-2017). *Mplus user's guide*. 8th ed. Muthén & Muthén, Los Angeles, CA.
- [19] M. S. Peteranetz, A. E. Flanigan, D. F. Shell, and L.-K. Soh. 2018. Future-oriented motivation and retention in computer science. In *SIGCSE '18: 49th ACM Technical Symposium on Computing Science Education*, Feb. 21–24, 2018, Baltimore, MD, USA.
- [20] K. G. Nelson, D. F. Shell, J. Husman, E. J. Fishman, and L.-K. Soh. 2015. Motivational and self-regulated learning profiles of students taking a foundational engineering course. *Journal of Engineering Education*, 104(1), 74-100.
- [21] L. Ott, B. Bettin, and L. Ureel. 2018, July. The impact of placement in introductory computer science courses on student persistence in a computing major. In *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education* (pp. 296-301). ACM.
- [22] M. Overmars. 2004. Teaching Computer Science through Game Design, *Computer*, 37(4), 81-83
- [23] M. S. Peteranetz, and L.-K. Soh. 2019. Predicting retention in undergraduate computer science courses: Perceived instrumentality and attendance. Presented at the annual meeting of the American Educational Research Association, Toronto, Ontario, Canada.
- [24] L. Porter, M. Guzdial, C. McDowell, and B. Simon. 2013. Success in introductory programming: What works?. *Communications of the ACM*, 56(8), 34-36.
- [25] K. Pulvers, and G. M. Diekhoff. 1999. The relationship between academic dishonesty and college classroom environments. *Research in Higher Education*, 40(4), 487-498.
- [26] E. Seymour and N. M. Hewitt. 1997. *Talking about leaving: Why undergraduates leave the sciences*. Westview Press, Boulder, CO.
- [27] D. F. Shell, M. P. Hazley, L.-K. Soh, E. Ingraham, and S. Ramsay. 2013, October. Associations of students' creativity, motivation, and self-regulation with learning and achievement in college computer science courses. In *Proceedings of the 2013 IEEE Frontiers in Education Conference (FIE)* (pp. 1637-1643). IEEE.
- [28] J. P. Somervell. 2006, June. Pair Programming: Not for Everyone? In *Proceedings of the International Conference on Frontiers in Education: Computer Science and Computer Engineering*, (pp. 303-307).
- [29] K. Empathy and A. D. Ritzhaupt. 2017. A meta-analysis of pair-programming in computer programming courses: Implications for educational practice. *ACM Transactions on Computing Education (TOCE)*, 17(4), 16.
- [30] J. Watkins and E. Mazur. 2013. Retaining students in science, technology, engineering, and mathematics (STEM) majors. *Journal of College Science Teaching*, 42(5), 36–41.
- [31] Z. S. Wilson, et al. 2011. Hierarchical mentoring: A transformative strategy for improving diversity and retention in undergraduate STEM disciplines. *Journal of Science Education and Technology*, 21(1), 148-156.
- [32] M. R. Young. 2005. The motivational effects of the classroom environment in facilitating self-regulated learning. *Journal of Marketing Education*, 27(1), 25- 40.