Towards More Gender Diversity in CS through an Artificial Intelligence Summer Program for High School Girls

Marie E. Vachovsky*
Stanford University
Stanford, U.S.A.
mvachovs@stanford.edu

Grace Wu*
Stanford University
Stanford, U.S.A.
gracew2@stanford.edu

Sorathan Chaturapruek*
Stanford University
Stanford, U.S.A.
sorathan@cs.stanford.edu

Olga Russakovsky*
Stanford University
Stanford, U.S.A.
olga@cs.stanford.edu

Richard Sommer
Stanford University
Stanford, U.S.A.
sommer@stanford.edu

Li Fei-Fei
Stanford University
Stanford, U.S.A.
feifeili@cs.stanford.edu

ABSTRACT

The field of computer science suffers from a lack of diversity. The Stanford Artificial Intelligence Laboratory's Outreach Summer (SAILORS), a two-week non-residential free summer program, recruits high school girls to computer science, specifically to Artificial Intelligence (AI). The program was organized by graduate student and professor volunteers. The goals of the pilot program are to increase interest in AI, contextualize technically rigorous AI concepts through societal impact, and address barriers that could discourage 10th grade girls from pursuing computer science. In this paper we describe the curriculum designed to achieve these goals. Survey results show students had a statistically significant increase in technical knowledge, interest in pursuing careers in AI, and confidence in succeeding in AI and computer science. Additionally, survey results show that the majority of the students found new role models, faculty support, and a sense of community in AI and computer science.

Keywords

Artificial Intelligence, K-12 Education, Outreach, Gender and Diversity, Summer Camps, Recruiting Women, Computer Science Education

1. INTRODUCTION

Recent diversity reports demonstrate a large gap between the percentage of women holding computing jobs compared to the percentage of men [1]. This disparity pervades in academia as well: in 2011, only 17.6% of Bachelor's degrees and 20.2% of Doctoral degrees in computer and information sciences from U.S. institutions were awarded to women [2].

*Equal contributions

SIGCSE '16, March 02 - 05, 2016, Memphis, TN, USA

ISBN 978-1-4503-3685-7/16/03...$15.00

DOI: http://dx.doi.org/10.1145/2839509.2844620

In this paper, we evaluate the Stanford Artificial Intelligence Laboratory's Outreach Summer (SAILORS) as a way of addressing the lack of diversity in Artificial Intelligence (AI). SAILORS recognizes that the AI field needs greater diversity, and sees AI's potential for technical and social impact education as a way to attract more women to the field of computer science (CS) as a whole. This potential has been untapped in high school CS education. SAILORS is the first AI camp for high school students that gives them a large-scope overview of the humanitarian applications of the field as well as the technical methods behind these applications. We evaluate the pilot program, a two-week nonresidential free[1] program run by 40+ graduate student and professor volunteers. The program focuses on gender diversity by targeting rising 10th grade girls, and has the following goals: 1) increase interest in AI among the students; 2) contextualize AI technical concepts through societal impact; and 3) address barriers that could discourage 10th grade girls from pursuing CS in the future.

Increase interest in AI. SAILORS exposes students to AI topics such as natural language processing (NLP) and computer vision, introduces these topics through technical skills, and gives them the opportunity to have close interactions with faculty and students from the Stanford AI Lab. Through these topics and opportunities, we aim to spark the students' curiosity and motivation.

Contextualize AI through social impact. SAILORS balances the technical skills with their application and frames technical skills as a stepping stone towards a larger goal. Within SAILORS curriculum, these problems have a humanitarian focus, which shows an alternative to the types of problems the stereotypical computer scientist works on. For example, girls chose from research projects that each applied AI technical skills towards a humanitarian purpose: making hospitals safer, assisting disaster relief, identifying cancer cells through the genome, and making driving safer (see section 4.5).

Address barriers for girls in computer science. SAILORS focuses on the gender-gap problem of recruitment within AI.

[1]Funded by sponsors

The program addresses three of the barriers that women in CS face at the undergraduate level, so that when SAILORS alumni reach university level, they will be equipped to handle these barriers. SAILORS addresses 1) lack of exposure to technical concepts early on [3, 4]; 2) limited perception of CS as a field [3, 5, 6]; and 3) lack of faculty and same-sex peer-to-peer support [3, 6, 7].

2. RELATED WORKS

Based on our literature review, SAILORS is unique in three ways: 1) it is the only summer AI program in the U.S. geared specifically towards high school girls; 2) it is the only high school summer program in the U.S. that teaches multiple subfields of AI (instead of focusing on only one, such as robotics [8] or bioinformatics [9]); and 3) it is the only summer program to teach these subfields through technically rigorous and socially relevant research projects.

CS programs for girls. There exist many out-of-school programs that target the same demographic as SAILORS; some of these work or worked towards similar goals. Girls on the Go was a week-long residential summer program held at Miami University for female high school sophomores and juniors [10]. Related to our goals, the camp introduced students to computing by exposing them to fields computer scientists collaborate with. Specifically, Girls on the Go taught girls to create an app that allows zookeepers to record animal behavior. The University of Delaware held an eight week half-day summer program for high school girls called Girls' POWER [11]. The camp's structure was similar to SAILORS's: the teaching was done by CS professors, the camp introduced girls to role models in CS research and industry, and there was a programming exercise after lectures.

AI programs for minorities. Some camps targeting underrepresented groups focused on one aspect of AI. Young Women in Computing (YWIC) was a five-week long summer program at New Mexico State University that targeted female high-school students [9]. Like SAILORS, YWIC aimed to increase interest, peer mentoring, and role models in CS. The program included a two-week long bioinformatics session, where students completed research projects in the field, e.g. investigated genes associated with particular illnesses. Additionally, the University of Minnesota Center for Distributed Robotics and the Digital Technology Center conducted a week-long day camp geared towards middle-school students underrepresented in CS (girls, African-Americans, Latinos) [8]. The camp, like SAILORS, was run by graduate student volunteers, and also found success in offering students real laboratory experience through robotics lessons. However, robotics and bioinformatics are only two topics in the larger field of AI. SAILORS aims to introduce students to the foundational technical concepts in AI, such as classifiers, and include topics from many concentrations in the field.

3. PROGRAM RECRUITMENT

We recruited students to apply for the program primarily through email. We emailed high school teachers in the bay area as well as students within the Stanford summer camp database (i.e. students that had applied for a Stanford camp before). As a pilot program, we found this method necessary to directly reach interested students. We welcomed students from all income levels and backgrounds to apply. Interested students completed a free online application composed of short response questions about their previous activities and classes, a longer question about their understanding of AI, and a teacher recommendation form. Students submitted a transcript. By the application deadline, we received 213 applications for 24 spots in the program. Overall, the following criteria were used throughout the admissions process: high academic performance in math and science classes, demonstrated leadership ability, enthusiasm for AI as exhibited through responses, strong teacher recommendations, and critical thinking as exhibited through the essay. Although previous CS experience was not a factor in admissions, all admitted students had some previous experience. Experience ranged from self-taught to AP CS; about half of the students taught themselves basic CS and 20% took a CS class in school (less than or equivalent to CS1). Our pool of admitted students consisted of girls passionate about technology who performed strongly in academics, but did not necessarily have much exposure to AI. All admitted students had high GPAs and many participated in science fairs and other extracurricular activities. A majority of them were Asian, reflecting our applicant pool, which was roughly 75% Asian. Additionally, all but two of the students came from California, as SAILORS is a non-residential program.

4. PROGRAM CURRICULUM

The two-week schedule was designed to fulfill the overall goals of SAILORS, and introduced girls to concepts in AI through a variety of engaging avenues. The first half of the daily schedule introduced students to research topics in AI and CS while the second half of the day allowed students to have first hand experience with some of the topics. Each activity targeted a combination of SAILORS's three goals, and balanced technological theory/rigor and social impact through AI. This balance is a part of what strongly differentiates SAILORS's curriculum from other CS camps. Because most students came in with a wide range of programming experience, the curriculum was designed assuming they had little to no programming knowledge.

4.1 Professor Lectures

Each morning began with breakfast with a professor, followed by an hour-long professor lecture. Students enjoyed the chance to speak personally with the professor about their research before the formal lecture. After breakfast, professors presented on the main topics and applications of their research. Topics included computer vision, cognition, biomedical informatics, robotics, information networks, natural language processing, human-robot interactions, computational sustainability, and aeronautics/astronautics. The lectures emphasized real-world applications to excite and inform the students about the types of problems AI and CS can solve. For example, one professor lectured on air controllers, and how he used dynamic programming to reduce air traffic collisions while also decreasing unnecessary flight path changes. Post-lecture breaks gave students an opportunity to interact closely with the professor, so that they may be inspired to learn more about the topic after the lecture.

4.2 Classroom Sessions

Following the professor lecture, graduate students explained in detail one technical component of the lecture. The graduate student instructors began with a lecture-style explanation; the students then worked on relevant problems to practice the concept. For example, one postdoctoral fellow gave a talk on Fisher's exact test. Students then used real data from a computational biology research site to compare genes based on varying parameters. For another session, students learned about inductive reasoning: when given a partial pattern of numbers, they guessed numbers that should belong, and compared the probability of their guesses to probabilistic programs that do the same. Other topics included nearest neighbor algorithm, correlation, graph search algorithms, and PID controllers. The classroom sessions gave students the confidence that they themselves can use the technical methods used in AI, and that they have the ability to delve into the technical aspects of the field.

4.3 Personal Growth Sessions

The personal growth sessions aimed to give girls the skills they need to be successful as women working in STEM. These skills were not AI specific and gave students the ability to thrive in the face of the more general challenges—microaggressions, working in teams, etc.—of both university level classes and of the AI/STEM workplace. The sessions were 1) talk with AI women; 2) team dynamics; 3) scientific communication; 4) industry panel; 5) design thinking; 6) how to recognize and react to microagressions; 7) time management; and 8) staying involved in AI/CS.

Excluding the panels, 1) and 4), the sessions were led in a seminar setting by a professional in the area. For example, a professor from Stanford Design school led the design thinking session. The sessions transitioned between short lectures, facilitated discussions, and interactive activities. The panels were conducted Q&A style, and the students asked relevant questions to the panelists.

4.4 Social Hour

Social hour gave the students a space to build friendships and community outside of their project work. Two social hours were spent exploring technology on campus: self-driving car and haptic surgical simulation; the remainder were unstructured.

4.5 Research Projects

A two hour time block each day was dedicated to research projects. The students were split up into four groups of six, with two graduate students leading each team. Project topics included computer vision, robotics, NLP, and computational biology; the students ranked the topics by preference and worked on the same project for the entire program. The project topics were carefully selected to be engaging, to emphasize the real-world impact of AI, to be technically stimulating, and to showcase the variety and depth of problems AI can solve. Additionally, the research projects were designed to closely mimic current AI research. Students could thus get a first-hand experience of what researchers in the field work on, and gain confidence that they can work on the same. In this way, the projects aimed to increase students' understanding of the technical methods used in AI as well as their excitement towards research in the field.

The coding for all of the projects was done in Python through the programming environment Jupyter. Each project included a short general introduction to coding. During the projects, the students implemented their code individually but worked in pairs to brainstorm ideas.

4.5.1 Making Hospitals Safer with Computer Vision

Students learned how to develop computer vision algorithms that can automatically understand actions in videos in order to monitor hand washing in hospitals. Their objective was to develop an algorithm that can recognize whether or not a doctor uses the hand sanitizing dispenser before he/she enters a room. First, to consturct a dataset, the students recorded a video and labeled each frame as having used the sanitizer or not. Then, instructors presented an overview of machine learning classifiers and introduced the background subtraction algorithm.

The students worked in pairs to code this algorithm. The instructors provided simple skeleton code, guidance when needed, and the evaluation code with the F-score metric; students largely coded the background subtraction algorithm independently. Manually choosing a threshold was the most technically rigorous portion. The students experimented with the size and location of the crop region for the algorithm as well as what to threshold (color channels, image difference, etc). After some manual trial and error, the students learned how to use the machine-learning assessment technique cross validation to automatically choose the thresholds.

In the remaining days, the students learned about and programmed a linear classifier. Because of limited time, the whole group with the instructors collectively coded the classifier on the board. Then, the students implemented it separately and compared the F-score of the background subtraction algorithm and the linear classifier. They discovered that the linear classifier performed much better than background subtraction. The students left with a broad understanding of the full pipeline of how to train and deploy an activity recognition model for a real-world healthcare application.

4.5.2 Assisting Disaster Relief with Natural Language Processing (NLP)

Students learned how to use NLP to determine which area of the world needs disaster relief, and which relief is needed (such as food, water, medical care). Students took a real corpus of tweets made during Hurricane Sandy, and automatically determined which of the tweets were relief requests for food. Each student built their own Naive Bayes classifier to classify the tweets.

Students began by exploring the raw data of tweets. Using the data, they created a simple rule-based classifier by writing the rules function in the classifier starter code. The students saw that this classifier does not generalize well, and, as an alternative, the instructors introduced Naive Bayes as a statistical model. For a hands-on demo, instructors used plastic cups to represent classes, grapes to represent instances, and grape color to represent features.

Students first used the Naive Bayes algorithm hands-on to classify Set cards by color, pattern, and number of icons. Using instructor starter code, they coded their algorithm. They then revisited the tweets, and explored the pros and cons of Naive Bayes as applied to tweets. Students wrote the code to extract features from a tweet, calculate class probabilities, calculate conditional probabilities, and predict the class of an unseen tweet. Students ran their final programs on the data, and compared the results to the rule-based classifier. At the end of the two weeks, the students saw that they have the ability to create something that processes real data to have a social impact.

4.5.3 Decoding DNA: Finding Meaning in the Genome

Students learned some of the technical methods used in computational biology. Students focused on gene expression data from several types of cancer, and used different classifiers to distinguish between normal and tumor samples, as well as between cancer types and stages of progression.

Students first learned the high-level concept of classifiers. As a hands-on example, students were given training grapes and tomatoes and developed features to distinguish them. This example intuitively demonstrated the k-nearest neighbors algorithm. After gaining the conceptual knowledge, students explored the online dataset of genes and the performances of k-nearest neighbor algorithm on various k (with working code written by the instructors). Then, students programmed the k-nearest neighbor algorithm themselves, working from instructor-written starter code. Students tested the performance of their algorithm on various datasets (provided by the instructors) with various parameters.

Student pairs brainstormed their own final experiments. One group tested the effect of random feature selection with k-nearest neighbors classification, while another tested the effects of balanced versus unbalanced datasets. By the end of the two weeks, students understood the basic biology of cancer, as well as several computational techniques used for analyzing cancer datasets to identify and classify tumors.

The Python code of the project, including the student's code, can be viewed online at goo.gl/okTkdj.

4.5.4 Self-Driving Cars and the Future of Personal Transportation

Students implemented a basic control system for a compact robotic car to navigate a system of roads represented by lines. Each day the instructors first presented a concept used by real autonomous cars, and then translated the concept to their robots, Pololu m3pi. The instructors taught three levels of control: 1) low level trajectory tracking, 2) motion planning, 3) routing.

In the first week, students learned control for low level trajectory tracking. The students focused on the algorithms, while communication infrastructure, data structures, etc. were prepared by the instructors. More specifically, the students programmed a proportional-integral-derivative (PID) controller to make a compact robotic car follow lines on a small road network using light sensors. They then iteratively tested the robots and adjusted control parameters and control strategies until the robot correctly and smoothly completed the task.

In the second week, students implemented Dijkstra's algorithm in Python to find the shortest path for the car. The students first independently wrote a function to return the cost of a path given the nodes of the path. The instructors then explained Dijkstra's algorithm, and the students worked in groups of two or three to implement Dijkstra's in code. The instructors provided a package to allow the students to see a visualization of their algorithm on a graph. Near the end of the week, students practiced route planning with other robotic cars on the line roads, simulating real roads. Students were able to receive information about the locations of other robotic cars and use Dijkstra's to choose a path to avoid collisions. In addition to technical knowledge, students gained an understanding of the potential value and societal impact of autonomous vehicles.

The Python code of the project, including the student's code, can be viewed online at goo.gl/kEWm3z.

4.6 Field Trips

The camp featured two field trips, one to the Computer History Museum with a tour led by Turing Award winner Ed Feigenbaum, and the other to Google to listen to a panel of Google female engineers and have lunch with the engineers. The first showed the students the rich history of computer science with an emphasis on AI, as told by the "father of expert systems". The second showed the students what an industry workplace looks like and what computer scientists in industry do on a day to day basis, with an emphasis on women in computer science. The students thoroughly enjoyed speaking with these professionals.

5. EVALUATION

We administered two sets of surveys. The first set included relevant questions from [12], as well as AI-oriented questions, and allowed us to establish a baseline for the attitudes and abilities of our admitted students. Relevant materials include [13, 14]. The second set of surveys focused on the admitted students pre and post participation in the 2 week summer program. Because the survey was AI specific, we did not use a validated instrument.

5.1 Baseline Survey

Students were asked questions about their attitudes towards math and science, their comfort level with peers at school, STEM career knowledge, humanitarian career interest, AI social interest, AI technical interest. For the analysis and for all following analysis, the Likert scale was quantified from 1 to 5, where 1 is the most negative and 5 is the most positive. For robustness of analysis, the questions were grouped based on the topic they measured. For each group, a scaled mean was calculated per participant. The resulting means suggest that the students have strong math and science backgrounds, and are passionate in engineering, humanitarian issues, and AI (mean above 4 in these areas). The survey also suggests students entered the program successful in general areas that SAILORS targets, for example STEM career knowledge and comfort in STEM classes (mean above 4 in these areas). For this reason, the second survey asks more specific questions about SAILORS's goals.

Before and after analysis was done on the baseline survey, with the same grouping as above. 11 paired t-tests were performed on 11 of the groups. Using the Benjamini-Hochberg procedure to control for false positive discovery rate, none of the t-tests were statistically significant. We speculate the high means did not leave much space for significant increase.

5.2 Pre/Post Survey

Students entered the program expressing a strong interest in CS and AI. At the end of the 2 week program, we found a statistically significant increase in their confidence and likelihood to pursue CS and AI. To test for a change in the students between the start and end of the program, paired analysis was done. The Likert scale questions were tested using a paired t-test. There were 11 such questions (Table 1). The two questions measuring interest in AI (Q13

Table 1: **Mean value changes in survey responses of the 24 SAILORS students before and after the program.**

#	Question	Mean Before	Mean After	p-value
12	How likely is it that you will have a career in AI in the future?	3.50	4.08	.0002
3	How likely do you think it is that you attain that position [your dream computer science position]?	3.33	3.83	.001
8	When I am learning CS, I know girls who I am comfortable asking for help.	3.08	3.92	.001
10	How likely is it that you will study AI?	3.79	4.25	.002
15	I can succeed in AI.	4.17	4.50	.002
9	How likely is it that you will major in computer science in the future?	4.29	4.58	.005
14	I can succeed in CS.	4.33	4.58	.011
11	How likely is it that you will have a career in CS in the future?	4.17	4.46	.016
Grouped 13,17	Grouped: 1) I want to learn more about AI. 2) I want to do research in AI.	4.71	4.75	.664
16	I want to do research in CS.	4.50	4.50	1.000

and Q17) were grouped together when tested, giving a total of 10 t-tests. The paired t-test analysis with Benjamini-Hochberg procedure found that questions 3, 8, 9, 10, 11, 12, 14, and 15 show a statistically significant ($\alpha = .05$) positive difference between pre-SAILORS student answers and post-SAILORS. Thus, there exists statistically significant evidence that SAILORS relates to 1) increase in confidence in CS (Q3 and Q14); 2) increase in confidence in AI (Q15); 3) increase in likelihood of pursuing CS (Q9 and Q11); 4) increase in likelihood of pursuing AI (Q10 and Q12); and 5) increase in same-sex peer-to-peer support (Q8).

The survey also asked technical questions to gauge the technical skills they learned at SAILORS (for example: "What is the main focus of controls engineering?"). The technical questions were tested using McNemar test performed as binomial exact test with Yates continuity correction. The test uses categories incorrect and correct, paired before and after. There is a statistically significant ($\alpha = .05$) change between answers for three out of five of the questions. For these three questions, we see an increase in the number of correct answers from before to after. These three questions addressed high-level technical concepts used in AI, as well as general CS, while the other two addressed simple probability and coding. Thus, there exists statistically significant evidence that SAILORS relates to an increase in CS and AI technical knowledge.

Students were asked if they had advisors in CS, role models in CS, and role models in AI (other than their parents). These questions were tested using McNemar test, with categories yes (student has advisor or role model) and no (does not have), paired before and after. There was a statistically significant ($\alpha = .05$) positive change between answers for AI role model ($p = .0002$). Thus, there exists statistically significant evidence that SAILORS relates to increase in AI role models. In the pre-survey, no student cited university level faculty as a CS advisor or role model. In the post-survey, 64% of those who have CS role models cited SAILORS-affiliated faculty, and 79% of those who have CS advisors cited SAILORS-affiliated faculty. Thus, there also exists statistically significant evidence that SAILORS relates to increase in faculty role models and faculty support.

When asked to give a short answer on what kind of problems people in AI work on, students first gave high-level answers along the lines of solving societal problems to make human lives easier. After the program, students gave more specific answers such as: "People in AI work on making models and graphs to predict or classify a range of items from diseases to birds." Additionally, when asked if their perception of who can be successful in CS changed and how, 83% of students responded yes, with the most cited responses being that anyone can be successful in CS, computer scientists can have other interests outside of CS, and other qualities besides academic strength (teamwork, problem solving) are also required to be successful in CS. Finally, we asked the students to elaborate on a problem in the world they are passionate about and what specific technical methods in AI can be used to solve that problem. Before the program, students cited "coding" and "robots". We saw an increase in the students' knowledge of which technical methods could be used to solve the problem. For example, one student stated that we can use the k-nearest neighbors algorithm to classify trash and recycling in order to prevent waste.

In addition to the questions above, we asked students for feedback on specific components of the curriculum. When asked to respond on Likert scale to the statements "The projects I built can help people" and "The projects I built were interesting", 95.8% of the students replied with strongly agree. 83.3% of students strongly agreed with the statement, "I am proud of what I built" and 75% of students strongly agreed with the statement "I am confident that I can work on a similar project in the future". For the personal growth sessions, when asked to respond to "There are people like me in AI", "There are people like me in CS fields", "I can find support in a CS field", and "I am comfortable speaking up for myself in a CS field", the students all responded with an average above 4 on the quantified Likert scale.

5.3 Limitations

Despite positive results, the pilot program had several limitations. First, the survey was unanonymized for internal purposes; future years should anonymize the survey. Second, the staff knew of the goals, and discussed some of the goals with the students, perhaps resulting in confirmation bias. Third, SAILORS did not reach low-income, African American, or Latino students. Since students of these backgrounds were not included in our study, we cannot draw conclusions from our results towards those demographics. Based on their applications, our students had access to other opportunities to learn about CS; while SAILORS may have been the only opportunity for them to learn AI, future years may target girls who do not have many technical opportunities.

6. CONCLUSION

The results show a positive impact of SAILORS upon the pilot-year students. Namely, the results support achieving goals 2 (contextualize technical AI concepts through social impact) and 3 (address barriers for girls in CS), and some support in achieving goal 1.

In support of goal 2, students' perception of AI and CS shifted to include a wider variety of people who can successfully work in the field. They also left knowing concrete, specific technical methods that can solve the problems they are passionate about. Research projects successfully allowed students to technically work on real-world applications.

In support of goal 3, students gained faculty support, same-sex peer-to-peer support, exposure to technical methods, and increased their AI and CS confidence. For faculty support, students gained faculty role models and advisors in CS and AI. For same-sex peer-to-peer support, students' comfort asking girls for help increased, and they cited SAILORS as a community. The personal growth sessions successfully allowed girls to relate to each other and other women in the field. The program has also created a community for SAILORS's volunteer faculty at the AI lab. For technical exposure, students were exposed to many technical methods used in AI; by the end of the program they had built classifiers, and correctly answered questions about technical methods used in AI. For confidence, students left feeling more likely to succeed in both AI and CS, as well as more likely to go into both fields. Students were also proud of their research projects, and their ability to engage real-world applicable technical methods.

The results do not directly show support for increase in interest. We speculate students already came in with a very high level of interest in AI (Likert scale mean 4.7/5) because of our admissions criteria. However, the increase in likelihood to go into AI suggests a potential increase in longer term interest.

All in all, SAILORS was successful this year in achieving its short-term goals. Our hope now is that the excitement and confidence the girls feel about AI and CS, as well as the community they have found, persist. To facilitate, we plan to hold events with both the students and AI faculty. Ultimately, SAILORS's pilot year has proven to be an effective way to educate students in AI and the societal impact of CS. In future years, we hope to reach a more diverse demographic of students as well as see the long term impact of the program.

7. ACKNOWLEDGEMENTS

We are grateful to all of the SAILORS staff, students, generous sponsors, and anonymous reviewers.

8. REFERENCES

[1] National Center for Women and Information Technology. Women and Information Technology By the Numbers, 2015. Retrieved August 21, 2015 from http://www.ncwit.org/sites/default/files/resources/btn_04032015_web.pdf.

[2] Degrees in computer and information sciences conferred by degree-granting institutions, by level of degree and sex of student: 1970-71 through 2010-11, 2012. Retrieved August 21, 2015 from http://nces.ed.gov/programs/digest/d12/tables/dt12_349.asp.

[3] Jane Margolis, Allan Fisher, and Faye Miller. The Anatomy of Interest: Women in Undergraduate Computer Science. *Women's Studies Quarterly*, 28(1/2):pp. 104–127, 2000.

[4] Greg Scragg and Jesse Smith. A Study of Barriers to Women in Undergraduate Computer Science. *SIGCSE Bull.*, 30(1):82–86, March 1998.

[5] Lori Carter. Why students with an Apparent Aptitude for Computer Science Don't Choose to Major in Computer Science. In *Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education*, SIGCSE '06, pages 27–31, New York, NY, USA, 2006. ACM.

[6] Neil Anderson, Colin Lankshear, Carolyn Timms, and Lyn Courtney. Because it's boring, irrelevant and I don't like computers: Why high school girls avoid professionally-oriented ICT subjects. *Computers & Education*, 50(4):1304 – 1318, 2008.

[7] J.M. Cohoon. Must there be so few? Including women in CS. In *Software Engineering, 2003. Proceedings. 25th International Conference on*, pages 668–674, May 2003.

[8] Kelly R. Cannon, Katherine A. Panciera, and Nikolaos P. Papanikolopoulos. Second annual robotics summer camp for underrepresented students. *SIGCSE Bull.*, 39(3):14–18, June 2007.

[9] Inna Pivkina, Enrico Pontelli, Rachel Jensen, and Jessica Haebe. Young women in computing: Lessons learned from an educational & outreach program. *SIGCSE Bull.*, 41(1):509–513, March 2009.

[10] Janet E. Burge, Gerald C. Gannod, Maureen Doyle, and Karen C. Davis. Girls on the Go: A CS summer camp to attract and inspire female high school students. In *Proceeding of the 44th ACM Technical Symposium on Computer Science Education*, SIGCSE '13, pages 615–620, New York, NY, USA, 2013. ACM.

[11] Lori Pollock, Kathleen McCoy, Sandra Carberry, Namratha Hundigopal, and Xiaoxin You. Increasing high school girls' self confidence and awareness of CS through a positive summer experience. *SIGCSE Bull.*, 36(1):185–189, March 2004.

[12] *Manual for Observing Patterns of Adaptive Learning (OPAL): A Protocol for Classroom Observations.*

[13] B. H. Loyd and C. P. Gressard. Computer attitude scale. *Journal of Computing Research.*, 15(3):241–259, June 1996.

[14] Brian Dorn and Allison Elliott Tew. Empirical validation and application of the computing attitudes survey. *Computer Science Education*, 25(1):1–36, 2015.