The Journal of Computing Sciences in Colleges

Papers of the 29th Annual CCSC Central Plains Conference

April 21st-22nd, 2023
Johnson County Community College
Overland Park, KS

Bin Peng, Associate Editor
Park University

Joseph Kendall-Morwick, Regional Editor
Washburn University

Volume 38, Number 6 **April 2023**

The Journal of Computing Sciences in Colleges (ISSN 1937-4771 print, 1937-4763 digital) is published at least six times per year and constitutes the refereed papers of regional conferences sponsored by the Consortium for Computing Sciences in Colleges.

Table of Contents

The Consortium for Computing Sciences in Colleges Board of Directors

Following is a listing of the contact information for the members of the Board of Directors and the Officers of the Consortium for Computing Sciences in Colleges (along with the years of expiration of their terms), as well as members serving CCSC:

bcdixon@csuchico.edu, Computer Science Department, California State University Chico, Chico, CA.

Serving the CCSC: These members are serving in positions as indicated:
Bin Peng, Associate Editor, bin.peng@park.edu, Department of Computer Science and Information Systems, Park University, 8700 NW River Park Drive, Parkville, MO 64152.
Ed Lindoo, Associate Treasurer & UPE Liaison, elindoo@regis.edu, Anderson College of Business and Computing, Regis University, 3333 Regis Boulevard, Denver, CO 80221.
George Dimitoglou, Comptroller, dimitoglou@hood.edu, Department of Computer Science, Hood college, 401 Rosemont Ave. Frederick, MD 21701.
Megan Thomas, Membership System Administrator, mthomas@cs.csustan.edu, Department of Computer Science, California State University Stanislaus, One University Circle, Turlock, CA 95382.
Karina Assiter, National Partners Chair, karinaassiter@landmark.edu.
Deborah Hwang, Webmaster, hwangdjh@acm.org.

CCSC National Partners

The Consortium is very happy to have the following as National Partners. If you have the opportunity please thank them for their support of computing in teaching institutions. As National Partners they are invited to participate in our regional conferences. Visit with their representatives there.

Platinum Level Partner

Google Cloud
GitHub
NSF – National Science Foundation

Gold Level Partner

zyBooks
Rephactor

Associate Level Partners

Mercury Learning and Information
Mercy College

Welcome to the 2023 CCSC Central Plains Conference

It is our pleasure to welcome you to the 29th annual Consortium for Computing Sciences in Colleges Central Plains Region Conference hosted by Johnson County Community College.

The conference this year boasts an appealing and diverse program. We are honored to have two distinguished speakers who are industry experts, Mr. Perry Copus from Garmin as the keynote speaker and Erin Christensen, KC Tech Council Chief Operating Officer as the banquet speaker. Additionally, the program will include a broad range of paper presentations, tutorials, workshops, lightning talks, nifty assignments, and a substantial collection of student research posters. This year we will be hosting our first Hack-A-Thon contest that should be enjoyable for everyone, as well.

The paper acceptance rate for this year was 58%. Moreover, each paper was reviewed by at least three reviewers. This ensures that the papers accepted in this program continue to be first-rate. We are certain that the conference program will benefit both computer science educators and students.

We are privileged to have worked with a group of dedicated individuals. Without the devoted committee members, reviewers, session moderators, and many other volunteers, this conference would not be achievable. We would also like to express our gratitude to the administration, staff, and colleagues in the Department of Computer Science and Information Systems at the Johnson County Community College who helped make this conference a success. Finally, thanks to the numerous individuals, vendors, and organizations whose support helped make the conference possible.

We are pleased to be hosting this year's conference in Overland Park, a beautiful city with many conveniences. And we are especially pleased to have the conference at the Johnson County Community College, which features a large and diverse student body.

We hope you find the conference both enlightening and enjoyable. We look forward to seeing you at the conference in April.

Mahmoud Yousef
University of Central Missouri
CCSC-2023 Central Plains Conference Chair

Perla Weaver
Johnson County Community College
CCSC-2023 Central Plains Conference Co-Chair

2023 CCSC Central Plains Conference Steering Committee

Conference Chair
Mahmoud Yousef .... University of Central Missouri
Conference Co-Chair
Perla Weaver .... Johnson County Community College
Conference Publicity
Michael P. Rogers .... University of Wisconsin Oshkosh
Mahmoud Yousef .... University of Central Missouri
Perla Weaver .... Johnson County Community College
Keynote Speakers
Mahmoud Yousef .... University of Central Missouri
Perla Weaver .... Johnson County Community College
Pre-Conference Workshop
Wen Hsin .... Park University
Judy Mullins .... Retired
Michael P. Rogers .... University of Wisconsin Oshkosh
Papers
Charles Riedesel .... University of Nebraska-Lincoln
Judy Mullins .... Retired
Ron McCleary .... Retired
Panels, Tutorials, Workshops
Ron McCleary .... Retired
Judy Mullins .... Retired
Mohammad Rawashdeh .... University of Central Missouri
Nifty Assignments
Michael P. Rogers .... University of Wisconsin Oshkosh
Brian Hare .... University of Missouri Kansas City
Bill Siever .... Washington University in St. Louis
Ron McCleary .... Retired
Judy Mullins .... Retired
Lightning Talks
Joseph Kendall-Morwick .... Washburn University
Bill Siever .... Washington University in St. Louis
K-12 Outreach
Belinda Copus .... University of Central Missouri
Perla Weaver .... Johnson County Community College
Diana Linville .... Northwest Missouri State University
K-12 Nifty Assignments and Lightning Talks
Bill Siever .... Washington University in St. Louis
Perla Weaver .... Johnson County Community College

Belinda Copus .... University of Central Missouri
Mohammad Rawashdeh .... University of Central Missouri

Student Paper Session

Scott Sigman .... Drury University
Wen Hsin .... Park University
Mahmoud Yousef .... University of Central Missouri

Student Poster Competition

Joseph Kendall-Morwick .... Washburn University
Ron McCleary .... Retired

Student Hack-a-thon

Scott Sigman .... Drury University
Chris Branton .... Drury University
Suzanne Smith .... Johnson County Community College
Bill Siever .... Washington University in St. Louis
Michael P. Rogers .... University of Wisconsin Oshkosh
Charles Riedesel .... University of Nebraska-Lincoln

Two-Year College Outreach

Suzanne Smith .... Johnson County Community College
Trisch Price .... Johnson County Community College
Mahmoud Yousef .... University of Central Missouri

Local Arrangements

Perla Weaver .... Johnson County Community College
Suzanne Smith .... Johnson County Community College
Trisch Price .... Johnson County Community College
Mahmoud Yousef .... University of Central Missouri

Regional Board — 2023 CCSC Central Plains Region

Reviewers — 2023 CCSC Central Plains Conference

Rad Alrifai ....Northeastern State University, Tahlequah, OK
Beth Arrowsmith ....University of Missouri - St. Louis, Saint Peters, MO
Ajay Bandi ....Northwest Missouri State University, Maryville, MO
Chris Branton ....Drury University, Springfield, MO
Kevin Brunner ....Graceland University, Lamoni, IA
John Buerck ....Saint Louis University, St. Louis, MO
David Bunde ....Knox College, Galesburg, IL
Karla Carter ....Bellevue University, Bellevue, NE
Chia-Chu Chiang ....University of Arkansas at Little Rock, Little Rock, AR
Belinda Copus ....University of Central Missouri, Warrensburg, MO
Anurag Dasgupta ....Valdosta State University, Hahira, GA
George Dimitoglou ....Hood College, Frederick, MD
Russell Feldhausen ....Kansas State University, Shawnee, KS
Aziz Fellah ....Northwest Missouri State University, Maryville, MO
Ernest Ferguson ....Northwest Missouri State University, Maryville, MO
David Furcy ....University of Wisconsin Oshkosh, Oshkosh, WI
Brian Hare ....University of Missouri-Kansas City, Kansas City, MO
David Heise ....Lincoln University, Jefferson University, MO
Suvineetha Herath ....Carl Sandburg College, Galesburg, IL
Wen Hsin ....Park University, Parkville, MO
Jeff Ifland ....State of Nebraska, Lincoln, NE
James Jones ....Graceland University, Lamoni, IA
Brian Kokensparger ....Creighton University, Omaha, NE
Srinivasarao Krishnaprasad ...Jacksonville State University, Jacksonville, AL
Diana Linville ....Northwest Missouri State University, Maryville, MO
Baochuan Lu ....Southwest Baptist University, Bolivar, MO
Eric Manley ....Drake University, Des Moines, IA
Chris Mayfield ....James Madison University, Harrisonburg, VA
Jose Metrolho ......Polytechnic Institute of Castelo Branco, Castelo Branco, Portugal
Judy Mullins ...University of Missouri-Kansas City (retired), St Joseph, MO
Kian Pokorny ....McKendree University, Lebanon, IL
Hassan Pournaghshband ....Kennesaw State University, Kennesaw, GA
Charles Riedesel ....University of Nebraska - Lincoln, Beatrice, NE
Michael Rogers ....University of Wisconsin Oshkosh, Oshkosh, WI
Jamil Saquer ....Missouri State University, Springfield, MO
William Siever ....Washington University, St. Louis, MO
Cindy Tu ....Northwest Missouri State University, Maryville, MO

In Memory of Richard Scott Bell (1972 – 2022)

Richard Scott Bell (Scott), Associate Professor at Northwest State University, passed away on Dec. 5, 2022.

Scott completed both his Bachelor of Science in Geological Engineering and Master of Science in Computer Science at the Missouri University of Science and Technology (a.k.a. UMR). Upon completion of his Master's degree, Scott found his passion for helping others learn while working as an instructor of Computer Information Systems at State Fair Community College in Jefferson City, MO (2000-2004) and then as an instructor of Information Technology at the University of Arkansas – Fort Smith (2006-2007). Scott joined Northwest as an instructor in Computer and Information Systems in 2007. Following a hiatus from 2010–2014 to complete his Ph.D. at Kansas State University, Scott returned to Northwest as an Assistant Professor. Scott's research and teaching interests included introductory computing courses, networking, and cybersecurity. Scott was especially enthusiastic about outreach activities that impacted K-12 students and educators.

In addition to his academic work, Scott was an avid outdoorsman, scuba diver, and storm chaser.

Scott was an active member of CCSC Central Plains from 2008 until 2019. In that time Scott wore many hats, including secretary; papers chair and co-chair; K-12 outreach chair; programming competition operations; 2-year school outreach chair; session moderator; and, most prominently, as site coordinator & conference chair in 2018. Scott was a leader, colleague, mentor, and friend to many in the CCSC Central Plains community.

Our deepest condolences to all who knew Scott, including his wife, father, and sister. He will be greatly missed.

The Engineering Generalist is [Probably] a Computer Scientist*

Friday Opening Keynote

Perry Copus
Technical Lead Engineer, Garmin Labs

Abstract

Computer Science education has more stakeholders and drivers than ever before: Increasing specializations (all apparently "crucial"), hiring-entity expectations for "work-ready" graduates, accrediting body demands, university retention emphases, parental return-on-investment calculi, program spin initiatives to lure gamers-cum-programmers, or artists-cum-game-programmers, or rebels-cum-white-hat-hackers... All vie to reshape the ever-evolving B.S.C.S. I don't want to talk about any of that. Let's have some fun instead. I will tell you about the most valuable person in the room on any team that is tackling complex, multidisciplinary engineering problems: The Engineering Generalist, whose peculiar capabilities bridge not only the various engineering disciplines, but also the technical, business, and user domains. As we define and verbally dissect this creature we will come to understand how the ideal fundamental training for such gems is not to be found in the hard engineering disciplines, nor in the mother-fields of the natural sciences, nor even in the grandmother-field of mathematics. It is rather the B.S.C.S. that stands as the veritable fundamentum, with a firm nod given to its historic, rigorous form.

Bio

Perry Copus is an entrepreneur, engineer, and technology generalist whose creative activities have spanned three decades and launched multiple technology companies and products. Copus founded Symmetric, a 3D graphics hardware company, in the 1990's and navigated its acquisition by a publicly-traded technology company. He also founded CoreIntellect, a venture-backed company with groundbreaking, proprietary natural language processing technology for filtering content at scale. His technology career has been eclectic and varied, with roles from software engineer to CEO, in companies from 12 to 50,000 employees, in industries from telecom to medical devices. His broader career includes university teaching in Computer Science and Cybersecurity, almost a decade serving as a Lutheran pastor, and various other roles. His formal education includes B.S. Computer Science, Master of Divinity, and M.S. Computer Science. He is a stage-IV cancer survivor, ultramarathon runner, gravel cyclist, and serial hobbyist.

Copus currently serves as Technical Lead Engineer at Garmin Labs, the advanced technology and product incubator for Garmin, a $5 billion consumer products company. In that role he helps advance Garmin's vision for future products as a multidisciplinary engineering leader and product architect. His role is broadly defined: derisking projects with key technical hurdles, investigating new technologies, realizing prototypes to prove extraordinary claims, and merging vision and technology to drive new products to the market.

Tech Specs: The Outlook of the Tech Industry and the Impact on the Regional Economy and Workforce*

Banquet Keynote

Erin Christensen
Chief Operating Officer, KC Tech Council

Abstract

As the voice of Kansas City's tech industry, the KC Tech Council is proud to elevate data to help us collectively find the opportunities to leverage our strengths, improve our deficiencies, and grow together as a tech hub of the future. Coming together as educators of future technologists within the region, you have an opportunity to engage in dialogue around trends in hiring practices and the direction of technology in the region.

Bio

Erin Christensen serves as the Chief Operating Officer of the KC Tech Council, a membership-based association serving as the regional advocate for Kansas City's tech industry. Prior to her appointment to COO in May 2022, Erin led workforce initiatives for the Tech Council which included the implementation of a leading tech apprenticeship program. Her role currently focuses on member and sponsor engagement, supporting workforce development, and member program development. Prior to joining the Tech Council in October 2020, Erin had a long tenure working in higher education with a recent focus on corporate engagement.

She currently serves on the BSIT Advisory Board for The University of Kansas Edwards Campus, i.c. Stars Kansas City Advisory Board, and on the Industry Executive Council for the North Kansas City School District. Erin is in her second year of the Greater Kansas City Chamber's Centurions Leadership Program. Receiving both her bachelor's and master's degrees from Northwest Missouri State University, Erin is originally from Rolla, MO but landed in Kansas City in 2014. She currently resides in Lees Summit, MO with her husband and daughter.

Influencing Technical Features of Using Podcast in E-learning*

Cindy Zhiling Tu[1] *and Gary Yu Zhao*[2]
[1]*School of Computer Science and Information Systems*
Northwest Missouri State University
Maryville, MO 64468
cindytu@nwmissouri.edu
[2]*Dakota State University*
Madison, SD 57042
gary.zhao@trojans.dsu.edu

Abstract

This study focuses on individuals' acceptance of podcasts in the e-learning context. Through a case study in a US mid-west university, this study explores the technical characteristics that affect instructors' and students' intention to adopt podcasts. Based on the Task-Technology Fit model, this study identifies the podcasts' influencing technical features such as the content type, length, background music, copyright clearance, and hosting service.

1 Introduction

A podcast is an audio/video presentation made available in digital format for download over the Internet. For example, an episodic series of digital audio or video files that a user can download to a personal device to listen to at a time of their choosing. Streaming applications and podcasting services provide a convenient and integrated way to manage a personal consumption queue across many podcast sources and playback devices. Podcasts are similar to radio programs in form, but they exist as audio files that can be played at a listener's convenience, anytime and anywhere. The number of persons who listen to podcasts continues to grow steadily. In 2022, there are at least 2,922,540 podcasts and 144,077,041 episodes in the world [2].

Podcast usage in e-learning is increasing dramatically, especially during the COVID-19 pandemic. Literature on e-learning in the last decade intensely suggests that podcasting initiatives have been on the rise across many nations [14, 10, 16, 9, 8]. Podcast usage helps institutions to serve their current students and to target those students who do not have the ability to attend regular classes. By adopting the podcast and changing it from an entertainment tool to a learning tool, educators can personalize and humanize e-Leaning by including rich media components in face-to-face, online, and hybrid classes in order to engage students in active, meaningful learning environments [13]. Podcasts enable students and teachers to share information without geographical or temporal limitations. Students can download the podcast of a recorded lesson for repeated learning at anytime from anywhere. It can be seen as an essential tool for communicating curriculum, assignments and information with students, parents, alumni, and the general community [12]. Moreover, the student-produced podcast can help students acquire new skills and improve their academic achievement because they are active participants in the fulfillment of the task, and they become a conductor of their own knowledge. Further, student-produced podcasts enhance engagement, competence in e-technologies, creativity, science communication skills and a broader understanding of the instructional content. It is obvious that podcasts can help both institutions and students improve their performance. However, the key to achieving success is when instructors and students adopt it.

Even the podcasts are easy to create and conveniently accessed from personal devices, and are free to use, most instructors and students do not use them as essential learning and teaching tool regularly. Why do people tend to use or refuse to use podcasts? What factors are determinants to affect the users' intention? These are interesting research topics. There is a lack of qualitative analysis that explores the technical characteristics of podcasts themselves and their role in influencing individual adoption in the e-learning context. Accordingly, this study aims to investigate the technical characteristics of the podcast affecting an individual's intention to use it with a grounded theoretical framework based on the Task-technology Fit (TTF) model.

2 Literature Review

2.1 Podcast and its technical characteristics

Podcasts are performances that are comprised of either audio or video MP3/MP4 recordings that can be downloaded directly to computers as well as to various mobile devices. The capability to transfer digital material to other portable devices provides an 'anytime, anywhere' media experience. Podcasts vary in style, format, and topical content. Podcasts are partially patterned on previous media genres but depart from them systematically in certain computationally observable stylistic respects [4]. Different from other information technology, podcasts have specific technology features that affect users' acceptance. Existing work has mentioned some relative features of podcasts. We synthesize and summarize seven technical features from the literature.

2.1.1 Types of the podcast

Podcast types are changing as new technologies, new types of content, and new use cases emerge. Five types of podcasting are based on the content and technology requirements. (1) Enhanced podcasts, also known as slide casts, are a type of podcast that combines audio with a slide show presentation. It is similar to a video podcast in that it combines dynamically generated imagery with audio synchronization, but it is different in that it uses presentation software to create the imagery and the sequence of display separately from the time of the original audio podcast recording [3]. Enhanced podcasts are widely used in businesses or in education. (2) A fiction podcast (also referred to as a "scripted podcast" or "narrative podcast") is similar to a radio drama, but in podcast form. They deliver a fictional story, usually told over multiple episodes and seasons, using multiple voice actors, dialogue, sound effects, and music to enrich the story [1]. Fiction podcasts cover a full range of literary genres from romance, comedy, and drama to fantasy, sci-fi, and detective fiction. Examples of fiction podcasts include The Bright Sessions, The Magnus Archives, Homecoming, Wooden Overcoats, We're Alive and Wolverine: The Long Night. (3) A podcast novel (also known as a "serialized audiobook" or "podcast audiobook") is a literary form that combines the concepts of a podcast and an audiobook. Like a traditional novel, a podcast novel is a work of literary fiction; however, it is recorded into episodes that are delivered online over a period. The episodes may be delivered automatically via RSS or through a website, blog, or other syndication methods. Episodes can be released on a regular schedule, e.g., once a week, or irregularly as each episode is completed. In the same manner as audiobooks, some podcast novels are elaborately narrated with sound effects and separate voice actors for each character, similar to a radio play or scripted podcast, but many have a single narrator and few or no sound effects. (4) A video podcast is a podcast that contains video content. Web television series are often distributed as video podcasts. Dead End Days, a serialized dark comedy about zombies released from October 31, 2003, through 2004, is commonly believed to be the first video podcast [6]. (5) Live podcasts. Many podcasts are recorded either in total or for specific episodes in front of a live audience. Ticket sales allow podcasters an additional way of monetizing. Some podcasts create specific live shows to tour, which are not necessarily included on the podcast feed.

The five types of podcasts used in the e-learning class can be slightly different from using them for general public podcasting. Enhanced podcasts are widely used for the class lecture. The instructor can use Microsoft PowerPoint to make the voice with the slides show. Fiction podcasts and podcast novels are generally used for arts, history, language, and literacy classes. Video podcasts are perfectly fit for lab classes, by which the instructor can show the lab operations step by step with video and audio. Finally, live podcasts can be used for class discussion/conversation/interview.

2.1.2 Audio quality

Producing quality audio can make the podcast appear more professional. It can also help improve the overall business model, appealing to paying advertisers. In the long term, quality recording equipment is a worthy investment in podcast success. It saves time in post-production, but most importantly, the better the podcast sounds, the more positively listeners will engage with the content. Editing software can help enhance audio quality, improving the listening experience for the audience. The editing process is also an opportunity to cut unnecessary content, allowing your broadcast to focus on a few key messages.

2.1.3 Background music

The podcast is in an audio-only format. The listeners don't see the creator's face and facial expressions or those podcasting guests, and there are no graphics bouncing around on the screen. Thus, the podcast should provide the listener with a full audio experience, which includes music. Good background music on the podcast helps set the tone, helps make transitions clear, adds entertainment value to the podcast, and creates brand recognition.

2.1.4 Length

The length of the podcast also depends on the subject, industry or genre. It could be that around 15-20 minutes perfect for the listeners. But maybe some specific audience wants more in-depth, exploratory information, in which case, 45-90 minutes would provide more value. Most research into educational podcasting advises sticking to 15 minutes maximum [15]. However, that figure was calculated in 2017. So, chances are high that the average podcast length is now less, with trends indicating that shorter podcasts are more popular.

2.1.5 Copyright clearance

The clearance of intellectual property rights is one of the critical success factors in e-learning. Both teachers and students likely would consider the podcast content that is clear of copyright infringement useful, though the teachers would bear the responsibility for clearing the copyright thereby assuming more risk than the students. Lin et al. (2013) investigated the relationship between copyright clearance and perceived ease of use, copyright clearance and behavioral intention based on the UTAUT model [12].

2.1.6 Hosting services

Podcast hosting services include storage space, outbound bandwidth, search websites, usage analytics, and distribution. For example, a popular free host for many educators is Anchor. Here is the hosting service provided by the supplier (https://themeisle.com/blog/best-free-podcast-hosting/).

Storage space: Unlimited
# of podcasts per account: 1
Outbound bandwidth: Unlimited
Website: ❌
Monetization: ✅ (US only)
Distribution: One-click distribution
Analytics: Advanced analytics built right into the Anchor dashboard

2.1.7 Structure

A good podcaster is intentional about how their show is structured, using it as a way to organize and express their ideas in a way that forms a meaningful story for their audience, even if it's simply a meaningful development in the conversation being had by the podcast's hosts.

2.2 Adoption of podcasts in education

Previous podcasting usage in the educational context focuses on the dissemination of information such as by broadcasting university news to staff and students, or by informing the new users of the library services [8]. As of the booming of e-learning, podcasting is widely adopted as a learning and teaching tool in class settings that includes campus, online, and hybrid classes, e.g., podcasting of guest lecture presentations, video podcasts for the lab class, live podcasting for the class discussion, etc. Podcasting is adopted to supplement class materials and support traditional mainstream e-learning. Such adoption could create a relationship that is based on continuous communication and interaction between teachers and students by having students engage in academic debate and in accessing timely academic research. Podcasting enables direct communication and interaction with students which go beyond the temporal and spatial limitations of conventional face-to-face education. Podcasting provides lecturers with the facility to emphasize the information, which they feel to be critical for the students, thus augmenting the teaching material. Moreover, the flexibility and affordability of podcasting cater to diverse students' needs by enabling repeated learning and offering an opportunity for the effective use of time.

Existing research examining instructors and students' acceptance of podcasts relied predominantly on TAM [5] and UTAUT [17] models. For example, Merhi (2015) investigated the technological, individual, and social aspects that influence the adoption of podcast use in education based on TAM and Diffusion of Innovation Theory [13]. Xigen Li & Li Zeng (2011) also used TAM and Diffusion of Innovation Theory to test the effects of both technology and non-technology factors on podcast adoption and use [11]. Lin et al. (2013) employed UTAUT as a base model to examine whether and how the teachers and students differentiate the podcast adoption patterns for educational purpose [12]. Ifedayo et al. (2021) investigated the factors that mediate the effect of lecturers' adoption of podcasts based on UTAUT model [10]. However, TAM and UTAUT only focus on users' beliefs and attitudes before or after adopting the new technology. Compared with models predicting adoption, the task-technology fit (TTF) (Goodhue & Thompson, 1995) model explains the acceptance of technology due to its characteristics and the fit to the task. Task-technology fit is the degree to which technology helps a user complete their tasks. Thus, Goodhue & Thompson (1995) suggest that the users intend to use the technology because they believe that they can improve their work performance by using the system if the functions of the technology correspond with their tasks [7]. The TAM and UTAUT do not concern the task, the technology, and the fit between the task and the technology, which is the focus of the TTF model. This study aims at examining the factors that influence an individual's intention to use podcasts in a class by focusing on podcasts' technical characteristics and the fit of task technology. Thus, the Task-Technology Fit (TTF) model is employed as a theoretical basis for our study.

3 Case Study

We conducted a case study in a mid-west university. In this case study, we select three undergraduate classes in the same course with 22 students, 14 students, and 20 students. In Class One and Class Two, teachers and students use various types of podcasts. Class Three does not use any podcasts at all. We designed different interview questions for instructors and students. We interviewed all three classes' instructors and 6 students in Class One, 4 students in Class Two, and 5 students in Class Three. Also, we conducted a survey for all students in 3 classes as a pilot study. From the survey, we got a good insight into how to design our interview questions and how to conduct the interview effectively and efficiently. We organized the documents and completed a thematic analysis after finishing all interviews. The results of the data analysis identified a few influencing technical features of the podcast, including the content type, length, background music, copyright clearance, and hosting service. These technical features positively and significantly influence Task-Technology Fit, which then drives users' intention to use podcasts in the e-learning classes. Users are more likely to use a technology if they perceive a better fit between technology and task [7]. Based on this view, TTF provides a theoretical basis to understand individual's acceptance of podcasts in the e-learning context with a focus on podcasts' characteristics and their fit to the tasks. By integrating the main concepts of TTF and our case study results, we propose a research model (Figure 1):

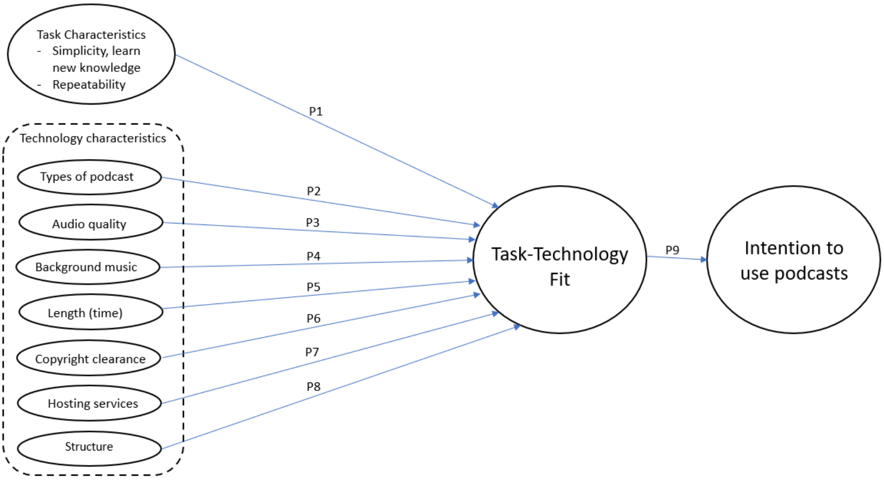


Figure 1: Proposed Research Model

We propose that:

P1: Task characteristics affects task-technology fit positively.

P2: Appropriate type of podcast affects task-technology fit positively.

P3: Audio quality has a positive relationship with the task-technology fit.

P4: Background music affects task-technology fit positively.
P5: Appropriate length affects task-technology fit positively.
P6: Copyright clearance affects task-technology fit positively.
P7: Hosting service affects task-technology fit positively.
P8: Podcast structure positively affect the task-technology fit.
P9: Task-technology fit influence users' intention to use podcasts positively.

4 Conclusion

This study addresses the podcast acceptance issue from the individual perspective. It is expected to contribute to both academics and practice. Theoretically, this study focuses on exploring the technical features of podcasts that influence the task-technology fit and then influence the intention of use in the classes, which so far has seldom been empirically studied in the literature and enriches the general TTF model. Practically, the results of this research will help instructors and students better understand individual users' behavior regarding using podcasts. Particularly, we get the rank of which podcast features influence the TTF, which can help us know how to improve technology characteristics and identify the top priority when the resource is limited.

References

[1] Fiction podcasts are giving rise to new audio storytellers (Voices). `https://www.voices.com/blog/fiction-podcasts/`. Accessed 2022-12-1.

[2] Podcast stats: How many podcasts are there? (Listen Notes). `https://www.listennotes.com/podcast-stats/`. Accessed 2022-12-1.

[3] Jordi Casteleyn and André Mottart. Slidecast yourself: Exploring the possibilities of a new online presentation tool. In *2010 IEEE International Professional Comunication Conference*, pages 255–261. IEEE, 2010.

[4] Ann Clifton, Sravana Reddy, Yongze Yu, Aasish Pappu, Rezvaneh Rezapour, Hamed Bonab, Maria Eskevich, Gareth Jones, Jussi Karlgren, Ben Carterette, et al. 100,000 podcasts: A spoken english document corpus. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5903–5917, 2020.

[5] Fred D Davis. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS quarterly*, pages 319–340, 1989.

[6] Jessica Ellis. What is a video podcast? (with pictures). https://www.easytechjunkie.com/what-is-a-video-podcast.htm. Accessed 2022-12-1.

[7] Dale L Goodhue and Ronald L Thompson. Task-technology fit and individual performance. *MIS quarterly*, pages 213–236, 1995.

[8] Howard Harris and Sungmin Park. Educational usages of podcasting. *British Journal of Educational Technology*, 39(3):548, 2008.

[9] Chien-Ta Bruce Ho, Yu-Hong Dennis Chou, and Hsuan-Yi Viola Fang. Technology adoption of podcast in language learning: Using taiwan and china as examples. *International Journal of e-Education, e-Business, e-Management and e-Learning*, 6(1):1, 2016.

[10] Adu Emmanuel Ifedayo, Azidah Abu Ziden, and Aziah Binti Ismail. Mediating effect of behavioural intention on podcast acceptance. *Education and information technologies*, 26:2767–2794, 2021.

[11] Xigen Li and Li Zeng. Technology attributes, perceived value of information, and social utility: Predicting podcast adoption and use. *Southwestern Mass Communication Journal*, 27(1), 2011.

[12] Shinjeng Lin, J Christopher Zimmer, and Velma Lee. Podcasting acceptance on campus: The differing perspectives of teachers and students. *Computers & Education*, 68:416–428, 2013.

[13] Mohammad I Merhi. Factors influencing higher education students to adopt podcast: An empirical study. *Computers & Education*, 83:32–43, 2015.

[14] Rafael J Rivera-Ortiz. Use of technology and podcast adoption among members of a higher education community: An institutional case study. *International Journal of Educational Excellence*, 7(2):31–57, 2021.

[15] Gilly Salmon and Palitha Edirisingha. *Podcasting for learning in universities*. McGraw-Hill Education (UK), 2008.

[16] Tal Samuel-Azran, Tal Laor, and Dana Tal. Who listens to podcasts, and why?: the israeli case. *Online information review*, 43(4):482–495, 2019.

[17] Viswanath Venkatesh, Michael G Morris, Gordon B Davis, and Fred D Davis. User acceptance of information technology: Toward a unified view. *MIS quarterly*, pages 425–478, 2003.

Introducing Virtual Reality to Undergraduate Students: A Hybrid Approach*

Tania Roy
Division of Natural Sciences
New College of Florida
Sarasota, FL 34243
troy@ncf.edu

Abstract

This paper documents an experience report on the design and implementation of an undergraduate course titled 'Introduction To Virtual Reality Systems' taught in two iterations: Spring 2021 and 2022. The paper describes the unique COVID-19 classroom restrictions and limited access to limited hardware. This elective course was aimed at second, third and fourth year students who had taken at least one programming course. The course focuses on the human-centered approach of designing VR experiences with emphasis on game design, software development, and social elements. The paper elaborates on course setup, teaching modalities, course content, sample assignments, and evaluation.

1 Introduction

Computer Science in a liberal arts institution allows for the possibility of teaching Interactive Systems related courses as electives. At New College Of Florida, we follow a similar sequence as recommended by ACM CC2013[13] with a core sequence of five required courses building the foundation for computational thinking and problem-solving. In our specific context, students who have taken the Introductory programming sequence of two programming courses (Python and Object-Oriented Programming) are eligible to take Elective courses in sub-areas such as Systems, Application, Theory, etc. This paper describes an approach to teaching Introduction to Virtual Reality Systems in an undergraduate institution as an Applications Elective. Two iterations of this course were taught and the evolution of the course design to accommodate student needs and teaching modalities have been described. Undergraduate institutions, especially small liberal arts institutions, do not always have the resources to teach these electives due to a lack of hardware and other computational resources. This paper discusses the approach to: a) teaching Virtual Reality (VR) to undergraduate students who have completed at least one programming language sequence b) low-cost resources that can assist small institutions to start teaching VR without compromising on learning outcomes c) assignments that build on software engineering paradigms and d) evaluation and feedback from students.

Undergraduate institutions/instructors who want to teach interactive computing and graphics often don't have the means to invest in thousands of dollars worth of technology. The pandemic forced us to think about how to make CS education more accessible in terms of affordability. Students taking remote courses do not always have access to computers/devices on-campus or a robust internet connection. This paper addresses some of those challenges and discusses modifications made to assignments to accommodate students without compromising on overall learning outcomes.

2 Background

VR development has been considered an upcoming area of critical need under the subcategory of Interactive Technologies; however, VR is rarely taught as an undergraduate course, or if it is - it's a senior elective cross-listed as a grad/undergrad course. VR development and problem-solving can also be used as a course where you build on introductory programming and problem-solving skills in addition to building on previous learning outcomes from courses.

VR has been taught in several institutions and their curriculum is available for review. VR has been added in 2020 as an area of critical need by the National Academy of Sciences and included in the CC2020 report published by ACM, however, it lacks learning outcomes and expectations of levels of competencies. Institutions that teach VR usually are R1 Ph.D. [3, 2, 1]. granting institutions with established researchers and research infrastructure to support hardware-intensive programs. The average cost of a VR headset is $400 and a development computer with compatible graphic processing units costs $1500-2000 which makes it a difficult investment for small institutions. Several graduates and undergraduate degree-granting CS programs teach VR reality development including peer liberal arts institutions such as Grinnell College[4]. The pedagogical philosophies and assignment suggestions are robust and provide a rich source of examples for developing a course. However, the primary struggle for small institutions is the cost of hardware. Grinnell College received a substantial startup fund to begin GCIEL and R1 institutions such as Clemson University's Virtual Environments Group have been able to receive several rounds of funding to accumulate hardware and hence allow a large group of grad/undergrad students to take these graphics-intensive courses. As a small public institution, New College Of Florida, did not have any VR/interactive computing hardware, hence we had to improvise and modify learning outcomes. This was additionally challenging due to COVID-19 some students were remote learning and sharing headsets was not recommended. The course was planned to focus primarily on connecting previous concepts to new knowledge domains.

3 Course Development and Planning

3.1 Strategy to offset the challenges of learning during COVID-19

Due to New College Of Florida's unique semester system, all Spring students spend 4 weeks on Independent Study Projects (ISPs). ISPs allow students to explore topics/learning material that interests them. This allowed for a 4-week rapid introduction and familiarity with C# before the students attended the VR class. The primary justification for doing a C# Bootcamp before VR was to help with: a) anxiety related to online learning and transitioning to a hybrid learning environment during Spring and b) provide additional practice for students who did not feel comfortable with Object Oriented Programming c) reduce the learning curve of adapting to new technology. All students taking Virtual Reality development were encouraged to attend this ISP. Course structure and assignments: Students followed along learning modules from a textbook Learning C# by Developing Games with Unity 2019 by Harrison Ferrone [9] along with daily learning activities, practice problems, bi-weekly programming assignments, and weekly quizzes. The students learned the fundamentals of programming in C# and explored game development mechanics on Unity using C#. The students were evaluated by weekly quizzes and programming assignments which involved a final individual project to demonstrate their competency.

3.2 Low-Cost Approach for VR Glasses

Due to the challenges of COVID-19, instruction for the Virtual Reality Development course had to be modified into a mix of in-person and virtual instruction mediums. Classes were held in person and on Zoom synchronously. As this was the first attempt at teaching VR in our institution, procuring multiple headsets that could be used by students in-person and remotely was challenging, hence the course was modified and most of the semester we pivoted towards learning Unity 3D [11] and developing simulations for Google Cardboard and an Android mobile device. Most students had mobile devices and were able to easily purchase Google cardboard devices [8] priced at $15-$20. VR courses that usually focus on technology-specific applications are not effective in the long run, as this field is evolving and hardware/software is becoming obsolete in less than 12 months. Instead, the approach of teaching computational thinking along with storytelling through technology nudges students to synthesize concepts from previous classes.

3.3 Grade Breakdown

a) Attendance & Participation - 5%
b) Programming Projects - 10% each (4 in total) which includes individual projects and group ones
c) Final Projects - 35% group based
d) Writing Assignments - 10% individual projects

3.4 Intended Learning Outcomes for Students

a) Understanding of Immersive Environments and the designing of interactive simulations in VR
b) Understanding the constraints related to designing for VR
c) Expertise with programming 3D assets and simulation development using Unity 3D game engine (prior experience required) for VR
d) Familiarity with VR: storytelling and gamification
e) Familiarity with VR and Applications in areas such as Education, Medical, and Entertainment
f) Developing VR simulations for low-cost (mobile VR) OR developing VR simulations for Headset based VR (Oculus)
g) Familiarity with how to test the effectiveness of VR simulations with users & storing data.

We had thirteen students complete the course over two iterations, who were all computer science majors. There were four first-year students, six second-year students, and three third-year students. For the whole semester in Spring 2021, two students attended remotely. However, due to COVID-related quarantine issues and suspected exposures, several students took the course remotely depending on what they were facing.

New College Of Florida has a low faculty-student ratio and having small enrollments in upper-level electives is not uncommon. We also had two students who withdrew before the end of the term. This course had a pre-requisite of familiarity with C# and Unity 3D (discussed in the section below). We had access to one Oculus Rift [7] headset, three Oculus Quest 2 headsets [5], and two Android phones for development along with one Windows and one Mac system to assist students. Most students were able to develop on their laptops for Google cardboard and co-developed for Oculus.

4 Course Assignments

For Spring 2021 (will also be referred to as Iteration 1), students completed three assignments on Google Cardboard-related assignments and had one coding exam and did two Oculus-related assignments. Students read papers on VR applications and presented the papers. For Spring 2022 (will also be referred to as Iteration 2), students completed two assignments related to Google Cardboard, two coding exams, and did two Oculus-related assignments along with a final project and demonstration sessions. Students read papers on VR applications and presented the papers. Most of these assignments were group projects and started in the classroom itself to allow students time to problem solve and start development in a structured environment. Figure 1 is a list of topics that were covered for both iterations. Two code review sessions were held in the second iteration vs. one in the first iteration. The code review was done when we introduced the Oculus Integration and Google Cardboard integration packages. The purpose of these was to familiarize the students with the packages and make development a less closed system (black box) as the enormity of these packages often overwhelm the student developers. In a group setting, students were responsible for identifying the connections and how they would use these packages for development purposes. This was very popular with students as they were able to unpack these integration packages, understand the coding practices of other developers, encounter issues with documentation, and additionally make connections with classes they were taking and how those principles were being used in a 3rd party package.

Students from Iteration 1 of the course, mentioned in their feedback that they were interested in the hardware components of the headset and the tracking software as they were unable to completely comprehend all the working components. Although this was a valid concern, Oculus does not share hardware schematics of the headset, so we improvised for the second iteration with teardown videos. A YouTube channel (iFixit) did a thorough breakdown of all the components, and students could understand the hardware components through this video. Although it is not an ideal solution, students received

		Topics covered 2022		***Topics covered 2021***
Week 1	No VR	Introduction to Immersive Systems	No VR	Introduction to Immersive Systems
Week 2	No VR	History of Virtual Reality & Motion Tracking & Controllers	No VR	History of Virtual Reality & Motion Tracking & Controllers
Week 3	Cardboard	Unity overview (recap), working with 3D environments, GitHub & Unity + Introduction to Google Cardboard	No VR	Unity overview (recap), working with 3D environments, GitHub & Unity
Week 4	Cardboard	The present and the future of XR & introduction to low-cost VR	Cardboard	The human behind the lenses
Week 5	Cardboard	The present and the future of XR & introduction to low-cost VR	Cardboard	The present and the future of XR & introduction to low-cost VR
Week 6	Cardboard	Perception, Action Loop & design guidelines to mitigate health challenges	Cardboard	Perception, Action Loop & design guidelines to mitigate health challenges
Week 7	Cardboard	Modeling the physical World in low-cost VR &: Camera tracking and 3D Rendering for Immersive Environments (low-cost VR)	Cardboard	Modeling the physical World in low-cost VR&: Camera tracking and 3D Rendering for Immersive
Week 8	Exam	Google Cardboard Review and Exam1	Cardboard	Presence of Agency and Interactivity
Week 9		SPRING BREAK	Cardboard	VR with tracking and controllers (Oculus)
Week 10	Oculus	VR Applications in Interactive Entertainment (Oculus) VR with tracking and controllers (Oculus)	Exam	Coding exam and transition week with 1 day off
Week 11	Oculus	VR Applications in Medical and Manufacturing (Oculus)	Oculus	VR Applications in Medical and Manufacturing (Oculus or low-cost VR)
Week 12	Oculus	VR for special target groups (how to design)	Oculus	VR for special target groups (how to design) and a brief introduction to user testing
Week 13	Exam	User Testing: Oculus exam and project proposal		** no classes
Week 14	Oculus	Final Project	Oculus	Final Project
Week 15	Oculus	Final Project presentation to the community	Oculus	Project presentation

Figure 1: Weekly breakdown of topics covered across 2 semesters

it well. We then supplemented knowledge related to the tracking algorithm through Hendrikson's paper [12] which elaborated on the SLAM algorithm. Figure 2 and Figure 3 show assignments given to students to accommodate students who have access to Oculus headsets vs. students who are doing the whole project on Google Cardboard.

4.1 Classroom Dynamics

4.2 Final Projects and Demonstration Session

For both iterations of the class, students were asked to complete a final project that utilized either the Oculus Quest 2 or Rift S or Google Cardboard-Android platform depending on access and availability. The role of the final project was to enable the students to create a portfolio artifact that merges the creativity and technical skills acquired throughout the semester. For iteration 2, the students demonstrated their projects to community members and all the simulations were enthusiastically received. The students received feedback on usability and mobility issues and picked 1-2 to amend before final submission along with documented code. The projects mostly were all gamified simulators with various gameplays such as treasure trails, mouse maze traps. Additionally, students chose to focus on environmental and mental health issues such as forest fire, stress-smash rooms, and educational simulations such as teaching students about logic gates and role of butterflies in pollination.

4.3 Student Evaluations

All the students for both iterations completed the course with satisfactory evaluation. At New College Of Florida we do not provide a numerical evaluation to students rather a narrative evaluation is provided along with a satisfactory vs. unsatisfactory designation [6]. There is no standard template for providing narrative feedback however most natural sciences faculty provide a combination of numerical and narrative evaluation. For students in the class, an overall numerical grade along with a description of their final project and skills acquired with the platform used is added to the narrative. Areas of success and improvements from the context of class participation, overall problem solving and programming skills are also added. A general comment on transferable skills is also added for the students to get a holistic overview of their performance in class. Narrative evaluations are uncommon in higher education and instructors adapting the curriculum can provide numerical grades fitting their educational needs.

Pick any of the projects: Oculus Devs

Please create this game to help DIY furniture developers make a chair

Guidelines - You are free to change it how you see fit but the simulation should have the following

OPTION 1 Oculus Devs

- Parts of a chair - 4 legs, 1 seat, 1 back is present in a room
- Design chart and instructions on how to put it together is present too
- Player can pick up a part and only the right rotation & part 'click' and the piece stays together
- Teleport to different parts of the room to pick up parts

OPTION 2
[Suitable for Google cardboard with incorporating 360-degree videos and modifying some of the tasks] & Oculus Devs

- Create a virtual Zen Garden (the scene can be imported from google poly or asset store)
- Add teleportation feature to allow people to move around in this space at constant speed.
- ***The garden has several activities (pick any 1 to complete for submission)***
  - Stack stones one on top of another and these stones are scattered around the space, they need to be picked up one at a time. Stones are also of varying sizes.
  - Make 3 stacks like the image below.
  - Collecting glowing lanterns and float them like balloons.
  - The simulations run from sunrise to sunset.
- Pet a kitten or a puppy and feed them treats. Puppy/kitten moves around, and you teleport to feed and pet.
- **The OVR hand-package integration or 3D hand model is used.**
- End of each task the ending screen shows that the person has attained Zen level.
- Ad**d calming music to the scene and positive reinforcement messages.**

OPTION 3

[suitable for Google cardboard] & Oculus Devs

Please create this game to help room designers

- A furniture warehouse with at-least 10 assets (5 for GC devs)
- Pick a room you want the player to design (teleportation or walking)(menu option for atleast 2 rooms)
- Allow player to customize room (Trigger-press)
- Provide hints/suggestions (Gaze interactions)
- Doors so that the player does not pop out of a wall
- Ending and starting screen for game.
- Teleport around the room to place furniture
- (Additional for GC devs) - Using google poly create your own assets

Figure 2: Options 1, 2, and 3 is a sample assignment for students to accommodate modality and technology access needs

Using Unity 3D and the Google Cardboard plugin please implement the following.

The basic concept & requirement list is provided below:

1) The simulation should be a scavenger hunt with 2 levels
- There should be a background appropriate for the storyline (360-degree image or simple backdrop)
- The player can look around in the environment and can interact with the object using GazeOverEvent & Cardboard.XR.Carboard.Api Trigger press.
    - At least 3 3D assets in each level
    - Text or audio feedback after successful interaction
    - Ability to leave the simulation in 3 mins if play is unsuccessful.
    - Total game play time : 2-3 mins
- It is up to the group to come up with a storyline for this scavenger hunt.
- GitHub File structure needs to be setup using the workshop notes/videos.

Exam Questions

1. [2.5 points] Import a scene into Unity, setup for google cardboard (example: Cube room or garden) [any free scene from Unity store or build your own similar to HelloCardBoard]
2. [5 points] Add 2 3D assets from Unity Store (example coffee & donut) [or free 3D asset from Unity store]
3. [2.5 points] Main camera is setup to allow head movement.
4. [5 points] Allowing the game to run for 60 secs then ending screen with text.
5. [5 points] Allowing assets to randomly float or stay stationary.
6. [5 points] Using gaze point controls to change coffee cup to an apple.
7. [2.5 points] - Duplicate Scene 1 and create Scene 2
8. [2.5 points] Using button click/trigger press to click on donut & change levels (Scene 1 to Scene 2)
    a. [10 points] Scene 2 is the same as Scene1 and here the user can use gazepoint and trigger press control to select coffee or donut & duplicate/ make a copy the element they selected.
    b. Allow the user to only duplicate upto 3 times (6 total) and then show a message in a textbox ' We are closed!'
9. [5points] When duplication happens some sort of animation is triggered too.
10. [5 points each] Extra credit
    a. Sound effects added.
    b. Game points appear every-time a new coffee or donut is added.

Figure 3: Coding Exam sample questions to ensure students learned the core concepts

Figure 4: Screenshots of final project game trailer videos.

4.4 Student Perspectives

Student feedback for both iterations was positive for the class. They mentioned that the regular hands-on projects that they worked on in class and outside helped them understand the material. They also enjoyed the creative freedom with development as the storylines were left up to them. Iteration 1 students struggled with the integral package-related challenges and did not enjoy how opaque Meta was with documentation. This led to the addition of code review of integration packages and the tear-down videos to supplement the lack of documentation and hardware schematics. Iteration 2 students reported similar feedback but also wanted to learn deployment skills which will be added to the next version of this class. All the students appreciated the 21-day C# and Unity boot camp ahead of the class as it reduced the anxiety of learning multiple new things and led us to focus on more transferable skills than syntax errors.

5 Retrospective

Teaching Virtual Reality Development in Spring 2021 during the pandemic with students both in-person and hybrid was a challenging pedagogical experience. However, this challenge lead to certain course design decisions which allow instructors to accommodate students with different requirements. Students did mention that they would have liked more hardware access (quantity and variety) and that issues to access to technology for courses similar to VR have been persistent since 1996 [10].

5.1 Focus on Transferable Skills

The hardware landscape of VR headsets (Oculus Go (May 2018), Rift (2016), Rift S (May 2019), Oculus Quest (May 2019) and Oculus Quest 2 (May 2020) and Quest 3 (expected 2023)) have been rapidly changing [5]. Given this fast evolution of learning, hardware-dependent development is counter-intuitive. So, the class was focused on learning problem-solving, game design, VR game design paradigms, and experience with a game engine (Unity3D) and language C#. This along with skills related to information literacy can be used by students in other areas of the work field [14].

5.2 Focus on Teamwork

Both iterations of the class focused on teamwork in class and outside, albeit the first iteration was socially distanced or over Zoom. To ensure that both students learned the material equitably, the introduction of in-class code review, coding exams, and project presentations helped with the assignments. This was also a good learning exercise as software development is done as a team activity. They all learned about the caveats on code collaborations and techniques for mitigating GitHub conflicts. This is also reflected in Zimmerman et al's [15] work where the authors reported similar findings with group work.

5.3 Future Work

Iteration 2 was developed on the student feedback of Iteration 1 and the easing of pandemic regulation enabled a more active learning environment. The introduction of code review (student code and integration packages) were very helpful and a learning experience for all along with the tear-down videos. For future iterations, we plan to focus a week on deployment, particularly to the Oculus store, which is restrictive due to licensing fees and copyright issues, however it will be a valuable experience if a simulated experience can be created. This course has been a popular offering for our students, leading to higher interest and requests for follow-up courses. We will continue to offer it however hardware challenges and access to affordable technology that evolves sustainably will be a challenge that we have to mitigate with time.

References

[1] CPSC 4110 - Virtual Reality Systems (Acalog ACMS). https://catalog.clemson.edu/preview_course_nopop.php?catoid=16&coid=65755. Accessed 2022-11-27.

[2] CSC475 Coursepage: Computer Science: UNCW. https://uncw.edu/csc/undergrad/courses/csc475.html. Accessed 2022-11-27.

[3] EE267: Virtual Reality. https://stanford.edu/class/ee267/. Accessed 2022-11-27.

[4] Grinnell College Immersive Experience Lab (GCIEL). https://www.grinnell.edu/academics/centers-programs/ctla/dlac/gciel. Accessed 2022-11-27.

[5] Meta Quest 2: Immersive All-In-One VR Headset. https://www.meta.com/quest/products/quest-2/. Accessed 2022-11-27.

[6] The model for the narrative evaluation system. https://www.ncf.edu/news/the-model-for-the-narrative-evaluation-system/. Accessed 2022-11-28.

[7] Oculus Rift S: PC-Powered VR Gaming Headset. https://www.oculus.com/rift-s/. Accessed 2022-11-28.

[8] Quickstart for Google Cardboard for Unity. https://developers.google.com/cardboard/develop/unity/quickstart. Accessed 2022-11-27.

[9] Unity 2019.1b. https://unity3d.com/unity/beta/2019.1. Accessed 2022-11-27.

[10] Douglas H Bell. Teaching virtual reality. *ACM SIGCSE Bulletin*, 28(2):56–61, 1996.

[11] Harrison Ferrone. *Learning C# by Developing games with unity 2019: Code in C# and build 3D games with unity.* Packt Publishing Ltd, 2019.

[12] Rorik Henrikson, Tovi Grossman, Sean Trowbridge, Daniel Wigdor, and Hrvoje Benko. Head-coupled kinematic template matching: A prediction model for ray pointing in VR. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–14, 2020.

[13] Mehran Sahami, Andrea Danyluk, Sally Fincher, Kathleen Fisher, D Grossman, Elizabeth Hawthorne, Randy Katz, Rich LeBlanc, Dave Reed, Steve Roach, et al. Computer science curricula 2013: Curriculum guidelines for undergraduate degree programs in computer science. *Association for Computing Machinery (ACM)-IEEE Computer Society*, 10:2534860, 2013.

[14] Sharon Stansfield. An introductory vr course for undergraduates incorporating foundation, experience and capstone. *ACM SIGCSE Bulletin*, 37(1):197–200, 2005.

[15] Guy W Zimmerman and Dena E Eber. When worlds collide! an interdisciplinary course in virtual-reality art. In *Proceedings of the thirty-second SIGCSE technical symposium on Computer Science Education*, pages 75–79, 2001.

Examining the Computational Thinking and Robotics Knowledge and Interest of Undergraduate Students in Two Teacher Education Courses*

Mohammad Q. Azhar[1], *Jillian Powers*[2], *Ann Musgrove*[2]
[1]*Computer Information Systems*
Borough of Manhattan Community College, CUNY
New York, NY 10007
`mazhar@bmcc.cuny.edu`
[2] *Department of Curriculum and Instruction*
Florida Atlantic University
Boca Raton, FL 33431
`{jrpowers, musgrove}@fau.edu`

Abstract

This research examines the impact of online instructional modules that integrated computational thinking (CT) and robotics in two teacher education courses on students' knowledge and interest in CT and robotics. Ninety-three (93) students from a lower-division instructional technology course and Fifty-nine (59) students from an upper-division instructional technology course participated in the study. One-sample paired t-tests were conducted to determine whether significant differences existed in participants' self-reported CT and robotics knowledge and interest from before and after the instruction. The results of the analysis found some statistically significant differences in mean CT and robotics knowledge and interest from pre-test to post-test in both courses suggesting this form of instruction can increase students' knowledge and interest in these concepts.

1 Introduction

Computational Thinking (CT) is an essential skill in the 21st-century workspace [5] as our world becomes increasingly digital and governed by data and artificial intelligence. CT is a problem-solving paradigm that uses logic, analysis, abstraction, and generalizing [17].

In this paper, we present our study that is aimed to examine the impact of an instructional module on the knowledge and interest in Computational thinking and robotics of undergraduate students in two teacher education courses. Our research involved 93 students from a lower-division and 59 students from an upper-division instructional technology course for teacher preparation at a large public university in Florida, U.S. over four semesters. This instructional module was collaboratively developed by computer science and education faculty that culminated with assignments in which students designed robots using various plugged in unplugged techniques. This paper presents the results of the study involving the following two research questions:

1. Is there a significant change in the undergraduate teacher education students' knowledge of CT and robotics after participating in instruction that integrated these concepts?

2. Is there a significant change in the undergraduate teacher education students' interest in CT and robotics after participating in instruction that integrated these concepts?

2 Related Work

Computer science educators are already well versed in the concept of CT; however, understanding of this field is less familiar to K12 educators despite the value it offers students [18]. CT's intrinsic interdisciplinary qualities are core to its value, but this can also make it challenging for pre-service teachers. Many teachers are uncertain about how to incorporate CT in the classroom [7]. An insufficient proportion of teacher preparation programs offer interventions for their students to hone CT skills [15]. Teacher training to include computing in elementary school education is lacking [12]. In response to this gap, courses are being developed at colleges of Engineering, Science, and Education employing various approaches to teach pre-service teachers how to embed CT and robotics into their future classrooms. The International Society for Technology in Education [1] updated student standards and educator competencies to further emphasize CT [2].

CT is increasingly explored in interventions in learning areas outside of computer science for K12 students [6]. For example, a course entitled "Toying with Technology" allows pre-service teachers studying at the School of Education at Iowa State University to apply CT through robotics tutorials and activities. The lessons culminated with student groups planning CT activities for kindergarten students at a partnering elementary school [16]. A study [19] in the field of educational psychology found pre-service teachers had a better grasp of incorporating and promoting abstraction, problem-solving, and algorithmic thinking in their future teaching after completing a module in CT. Findings indicate that robotics activities can be a powerful way to increase familiarity with CT for pre-service teachers [10]. [14] successfully integrated CT into the curriculum for pre-service and graduate teachers with a robot activity. Pre-service teachers' algorithmic thinking has been shown to improve with LEGO WeDo robotic activities. [4]. Flipped learning modules have shown success in undergraduate education classes using the block programming app Hopscotch [20]. Adler and Beck [3] demonstrated in a mixed-methods study that incorporating an introductory computer science course, "Computer Science for All," improved confidence and self-efficacy with CT in all students and increased education students' confidence to incorporate CT in their future classrooms. A study exploring pre-service teachers' perceptions of CT in K12 education found that those who engaged with hands-on application of CT in a makerspace were better equipped to envision how to integrate CT into the classroom than participants who did not engage with practical applications of CT [8]. A frequently utilized model to explore the intersection of technology and pedagogy is the Technological Pedagogical Content Knowledge (TPACK) model [11], which suggests incorporating CT into the K12 teacher education curriculum will further Pedagogical Content Knowledge (PCK). Findings indicate the need to integrate CT into coursework on teaching methods, as a curriculum that positively influences participant knowledge of CT may not go beyond a surface conceptualization of CT, failing to result in lessons designed to incorporate CT tools in a meaningful way [13].

3 Methods

3.1 Description of the Modules

The lessons created for this research were adapted from Robotics Unplugged, which was presented at the Grace Hopper Conference [9]. The researchers adjusted the activities that were originally designed for undergraduate computer science students for undergraduate teacher education students in lower and upper-division courses that took place online. In both the lower and upper-division teacher education courses, students completed two online modules encompassing CT, AI, and robotics in education. The key difference was the lower-division students created robots that could solve a problem in one of the following areas: health care, the environment, aging, or education. The upper-division students focused on robots as assistive technology tools. In both classes, the first lesson was focused on CT and robotics. This lesson started by sharing background information regarding the Computer Science for All initiative and a video titled Computational Thinking: A Digital Age Skill for Everyone [5]. Next, a video lecture on CT and robotics given by the education and computer science faculty together was posted along with an accompanying PowerPoint. In addition to these resources, the upper-division students were provided with assistive technology resources and examples of assistive robots. Both groups were instructed to use MS Word tools, including shapes, text boxes, and images, to draw and label the features and functions of the robot. The lower division students were asked to answer the following reflection questions about their robot:

1. What your robot service does, and why is this important to the societal area?
2. What features does your robot need (hardware, software, anything additional) to successfully perform its functions?
3. How does each sticky note you have placed on your robot relate to the features it needs?
4. What types of physical obstacles would your robot have to overcome?
5. What types of psychological challenges do people have to overcome to accept your robot service?

The upper-division students were asked to answer the same questions as the lower-division students with the addition of the following questions:

- What problem in education does your robot help solve?
- What features help with certain accessibility challenges?

3.2 Data Collection

Data were collected using pre-and post-test surveys developed with Qualtrics software. Separate surveys were created for the lower-division and upper-division courses. In each survey, the first group of items collected demographic and background information about the participants. To assess CT and robotics knowledge and interest, items from the instrument used by [19] were adapted for the context of undergraduate teacher education students. Specifically, participants were asked to rate their level of agreement with each of the items regarding CT and robotics knowledge, and interests were measured on a 7-point Likert scale ranging from strongly agree (7) to strongly disagree (1).

- CTK1 - Computational thinking involves thinking logically to solve problems
- CTK 2 - Computational thinking involves abstracting general principles and applying them to other situations
- RK1- Robotics involves the design of machines that can sense the world and act on it
- RK 2 - Robotics involves the design of machines that can make decisions based on computations
- CTK 1 I think computational thinking is boring
- CTII 2 - The challenge of designing computational thinking using robotics appeals to me "
- RI 1 - I think robotics is boring
- RI 2 - The challenge of designing robots using robotics appeals to me

The research commenced after approval from the university institutional review board was granted. Within the courses, students were provided with a link to the pre-test survey prior to completing the instruction and the post-test survey after the instruction. Although completion of the surveys was a required assignment in both courses, participation in the research was voluntary, and only students who consented to participate were included in this study.

3.3 Data Analysis

The researchers entered the data into SPSS software for analysis. Descriptive statistics for demographic and background variables were computed to describe the study's participants. The survey items regarding knowledge and interest in CT and robotics were also summarized by calculating descriptive statistics. One-sample paired t-tests were run to determine whether there was a statistically significant mean difference between self-reported ratings of knowledge and interest in CT and robotics of participants between the pre-test and post-test surveys. Differences in means of the paired values was calculated and plotted on histograms to assess the validity of the paired t-test results.

3.4 Participants

A total of 93 students from a lower-division instructional technology course and 59 students from an upper-division instructional technology course at a large public university in the southeastern United States participated in the study. All courses were part of an undergraduate teacher education program and took place from spring 2020 through spring 2021. Though one section of the lower-division course initially started as a face-to-face course. Due to the transition to online learning in the spring of 2020, all of the lessons that were a part of this study were delivered via online learning through Canvas. Across both courses, most students were pursuing a degree in the education field 129 (84.9%), including early childhood, elementary, secondary, and exceptional student education majors. As is common among education majors, the participants were predominately female, 130 (85.5%), 21 were male (13.9% and one participant preferred not to answer the survey item about gender (0.6%). Regarding the number of participants self-reporting taking a college-level computer science class, 10 (10.8%) responded "yes" and 83 (89.2%) "no" from the lower-division course, while 7 (11.9%) said "yes" and 53 (88.1%) said "no" from the upper-division course. This data suggests that overall, most of the participants from both courses did not have a lot of background knowledge in computer science prior to this lesson.

4 Results

4.1 Computational Thinking and Robotics Knowledge

As shown in Figure 1, the lower-division participants, the results of the paired t-test for the self-reported survey items RK 1 $t(89) = 2.931$, $p = 0.004$, $d = 0.444$, 95% CI [0.143, 0.746], and RK 2 $t(89) = 3.569$, $p = 0.001$, $d = 0.433$, 95% CI [0.192, 0.675] differed significantly from pre-test rating. Therefore, we can reject the null hypothesis that there is no difference in participants' mean pre- and post-ratings of these items (RK 1 and RK 2). This suggests that these two measures of robot knowledge increased for lower-division students who participated in the lesson. For the upper-division participants, paired t-test results for the self-reported survey item CTK 2 indicated that post-test ratings differed significantly from pre-test rating $t(57) = 2.258$, $p = 0.028$, $d = 0.296$, 95% CI [0.18, 1.154]. Therefore, we can reject the null hypothesis that there is no difference in participants' mean pre- and post-ratings of this item (CTK 2). This finding indicates that upper-division students' knowledge of CT as it related to abstracting and applying increased after completing the lesson. None of the items regarding lower-division CT knowledge or upper-division robotics knowledge differed significantly from pre-test to post-test.

4.2 Computational Thinking and Robotics Interest

The results of the paired t-test for participants' self-reported ratings of interest CT and robotics are presented in Figure 2. As shown in Table 2, lower-division

Survey Item		Paired Differences Mean	SD	Std. Error Mean	95% Confidence Interval of the Difference Lower	Upper	t	df	Sig. (2-tailed)
Lower-Division	CTK1	.033	1.178	.123	-.212	.278	.267	90	.790
	CTK 2	.033	1.206	.126	-.218	.284	.261	90	.795
	RK 1	.444	1.439	.152	.143	.746	2.931	89	.004*
	RK 2	.433	1.152	.121	.192	.675	3.569	89	.001*
Upper-Division	CTK 1	.034	.878	.115	-.196	.265	.299	57	.766
	CTK 2	.345	1.163	.153	.039	.651	2.258	57	.028*
	RK 1	-.017	1.481	.194	-.407	.372	.372	57	.930
	RK 2	-.155	1.295	.170	-.496	.185	.185	57	.365

Figure 1: Computational Thinking and Robotics Knowledge Paired Samples Tests

participants, the results of the paired t-test for the self-reported survey items CTI 2 $t(90) = -2.060$, $p = 0.042$, $d = -0.352$, 95% CI [0.012, 2.060] and RI $t(57) = -2.066$, $p = 0.043$, $d = -.311$, 95% CI [0.664, -.018]. For upper-division participants, the self-reported survey item CTI 1 differed significantly from pre-test to post-test $t(57) = -2.066$, $p = 0.043$, $d = -0.271$, 95% CI [-1.15, -.018]. Note that the mean decrease in R1 was negative, but this was desired because this measure of robotics interest asked whether students thought robotics was boring. Therefore, we can reject the null hypothesis that there is no difference in participants' mean pre- and post-ratings of this item. In addition, a statistically significant decrease in one measure of robotics knowledge, RI2 = -.103, $p = 0.006$ was found, indicating that upper-division students found the challenge of designing robots appeals to them less.

5 Discussion

This research examined the knowledge and interest in CT and robotics of undergraduate students in two teacher education courses. The lower-division course is typically taken in the early stages of the degree program, and the upper-division course at a later stage. Regarding CT and robotics knowledge, the findings indicated that participating in the module that integrated these concepts into instruction significantly influenced knowledge ratings of robotics for lower-division students and knowledge ratings of CT for upper-division students. Specifically, for lower-division participants, these included the survey items "robotics involves the design of machines that can sense the world and

Survey Item		Paired Differences Mean	SD	Std. Error Mean	95% Confidence Interval of the Difference Lower	Upper	t	df	Sig. (2-tailed)
Lower-Division	CTI 1	-.231	1.700	.178	-.585	-1.295	-1.295	90	.199
	CTI 2	.352	1.629	.171	.012	0.012	2.060	90	.042*
	RI 1	-.311	1.687	.178	-.664	-.018	-2.066	57	.043*
	RI 2	.433	1.469	.155	.126	.720	1.376	57	.174
Upper-Division	CTI 1	-.586	2.161	.284	-1.154	-.018	-2.066	57	.043*
	CTI 2	.293	1.622	.213	-.133	.720	1.376	57	.174
	RI 1	-.207	1.755	.230	-.668	.042	-1.750	89	.084
	RI 2	-.103	1.608	.211	-.526	.741	2.798	89	.006*

Figure 2: Computational Thinking and Robotics Interest Paired Samples Tests

act on it" and "robotics involves the design of machines that can make decisions based on computations." For upper-division participants, significant differences were found in one of the measures of CT we examined, "computational thinking involves abstracting general principles and applying them to other situations." For all three of these significant findings, the mean increased from pre- to post-test, indicating that self-reported knowledge increased. It should be emphasized that the two courses had two distinct instructional modules tailored specifically to integrate CT and robotics with course standards. Therefore, our discussion of findings across groups is purely observational but not scientific in nature. Our findings pertaining to interest in CT and robotics also demonstrated the items that yielded a significant difference between pre- and post-test surveys were mixed. For lower-division participants, the CT interest item "the challenge of designing computational thinking using robotics appeals to me "and the robotics interest item "robotics involves the design of machines that can sense the world and act on it" differed significantly between pre- to post-test. For these two significant findings, the mean increased from pre- to post-test, indicating that self-reported interest increased. For the upper-division participants, another CT interest item, " I think computational thinking is boring," yielded a statistically significant result. For this interest measure, mean self-reported interest decreased from pre- to post-test, which was the desired result because it suggested that the participants did not find CT to be as boring of a topic as they did before the lesson. However, one robotics interest item, "the challenge of designing robots using robotics appeals to me." decreased which was a disappointing finding. Perhaps this may relate to the finding that there were no significant changes in their robotics knowledge, and hence the lesson was less engaging for this group.

6 Conclusion and Future Work

This research contributes to the ongoing effort to develop innovative methods of integrating fundamental computer science topics into the teacher education curriculum. Together the study findings supported the notion that including CT and robotics in the instruction of pre-service teachers can positively shape their knowledge and interest in these concepts. The instructional modules examined in this study took place online, and the research was based on online survey data. In order to gain deeper insights into teacher education students' knowledge and interest in CT and robotics, future work might include collecting and analyzing interview data and systematically analyzing student artifacts.

Acknowledgements

Special thanks to Dr. Susan Imberman for the inspiration. Additional thanks to Dr. John Tsapogas, Ingrid Valentin, and Marisa Day for their support. This work was partially supported by CUNY Open Educational Resources (OER) and CUNY Faculty Travel Grant. All opinions reflected in this paper are those of the authors and not necessarily those of the CUNY OER and CUNY Travel Grant.

References

[1] ISTE 2016. `https://www.iste.org/explore/ISTE-blog/The-2016-ISTE-Standards-for-Students-are-here\%21`.

[2] ISTE 2018. `https://www.iste.org/standards/iste-standards-for-education-leaders`.

[3] Rachel F Adler and Kristan Beck. Developing an introductory computer science course for pre-service teachers. *Journal of Technology and Teacher Education*, 28(3):519–541, 2020.

[4] Charoula Angeli. The effects of scaffolded programming scripts on pre-service teachers' computational thinking: Developing algorithmic thinking through programming robots. *International Journal of Child-Computer Interaction*, 31:100329, 2022.

[5] David Barr, John Harrison, and Leslie Conery. Computational thinking: A digital age skill for everyone. *Learning & Leading with Technology*, 38(6):20–23, 2011.

[6] Nathan Bean, Joshua Weese, Russell Feldhausen, and R Scott Bell. Starting from scratch: Developing a pre-service teacher training program in computational thinking. In *2015 IEEE Frontiers in Education Conference (FIE)*, pages 1–8. IEEE, 2015.

[7] Yu-hui Chang and Lana Peterson. Pre-service teachers' perceptions of computational thinking. *Journal of Technology and Teacher Education*, 26(3):353–374, 2018.

[8] Ya-Wen Cheng, Pei-Chen Sun, and Nian-Shing Chen. An investigation of the needs on educational robots. In *2017 IEEE 17th International Conference on Advanced Learning Technologies (ICALT)*, pages 536–538. IEEE, 2017.

[9] Susan P Imberman, Jean McManus, and Gina Otts. Creating serious robots that improve society. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[10] Kamini Jaipal-Jamani and Charoula Angeli. Effect of robotics on elementary pre-service teachers' self-efficacy, science learning, and computational thinking. *Journal of Science Education and Technology*, 26(2):175–192, 2017.

[11] Matthew Koehler and Punya Mishra. What is technological pedagogical content knowledge (tpack)? *Contemporary issues in technology and teacher education*, 9(1):60–70, 2009.

[12] Stacie L Mason and Peter J Rich. Preparing elementary school teachers to teach computing, coding, and computational thinking. *Contemporary Issues in Technology and Teacher Education*, 19(4):790–824, 2019.

[13] Chrystalla Mouza, Hui Yang, Yi-Cheng Pan, Sule Yilmaz Ozden, and Lori Pollock. Resetting educational technology coursework for pre-service teachers: A computational thinking approach to the development of technological pedagogical content knowledge (tpack). *Australasian Journal of Educational Technology*, 33(3), 2017.

[14] Jillian Powers and Mohammad Azhar. Preparing teachers to engage students in computational thinking through an introductory robot design activity. *Journal of Computers in Mathematics and Science Teaching*, 39(1):49–70, 2020.

[15] Leslie Suters. Elementary preservice teacher coursework design for developing science and mathematics computational thinking practices. *Contemporary Issues in Technology and Teacher Education*, 21(2):360–440, 2021.

[16] Kristina Tank and Dennis Culver. A model for integrating computational thinking into an undergraduate preservice teacher education course. In *Society for Information Technology & Teacher Education International Conference*, pages 335–340. Association for the Advancement of Computing in Education (AACE), 2019.

[17] Jeannette M Wing. Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1881):3717–3725, 2008.

[18] Aman Yadav, Sarah Gretter, Jon Good, and Tamika McLean. Computational thinking in teacher education. In *Emerging research, practice, and policy on computational thinking*, pages 205–220. Springer, 2017.

[19] Aman Yadav, Hai Hong, and Chris Stephenson. Computational thinking for all: Pedagogical approaches to embedding 21st century problem solving in k-12 classrooms. *TechTrends*, 60(6):565–568, 2016.

[20] Shenghua Zha, Yi Jin, Pamela Moore, and Joe Gaston. Hopscotch into coding: introducing pre-service teachers computational thinking. *TechTrends*, 64(1):17–28, 2020.

Effect of Teaching a Course with a Remote Instructor*

Charles Hoot, Nathan W. Eloe, and Chandra Mouli Kotteti
School of Computer Science and Information Systems
Northwest Missouri State University
Maryville, MO 64468
{hoot,nathane,chandra}@nwmissouri.edu

Abstract

Quickly adapting to new modes of instruction during a global pandemic has changed how teachers deal with locality in education. Efficiently adapting new tools to account for where students and instructors are in relation to each other enabled educators to be more flexible in content delivery methods and evaluation. Unfortunately the events of the pandemic are still felt in the global community that is academia. This paper serves as an experience report on a situation in which a large class was co-taught by two faculty members, but one of them was unable to teach their sections of the class face to face. The effect of face to face versus remote delivery on both student performance and self-assessed preparedness is presented.

1 Introduction

Adapting to changing delivery methods during a global pandemic was an educational experience for a significant portion of individuals in academia. During the most restrictive times of the pandemic, a significant amount of research examined how best to deliver content on a wide scale for courses not initially designed for remote learning [4, 2]. While converting courses designed for face to face instruction was required at the time [1], the overall effect on education was not positive [3].

One relatively common aspect of teaching remotely during the pandemic was the physical separation of all parties involved, both students and teachers. Indeed, the need for the creation of "Classroom Community" in online classes has been assessed over time [5]. This physical separation of individuals can be a cause of decreased student engagement and success.

What then, are the effects on a class where the students are together in a classroom while the teacher presents material synchronously but remotely? How is the students' preparedness (both perceived and actual) impacted by such a situation? This paper is a report on a situation where the effects of this teaching configuration could be measured to some extent.

2 Experimental Setup

Due to travel restrictions, one of the two instructors for our Developing Web Applications and Services course was unable to return to campus at the beginning of the semester. Instead of switching to an asynchronous instruction methodology, the decision was made to have the instructor deliver lectures synchronously over Zoom to the students in their assigned classroom. It was hoped that having the students physically present would increase participation by reducing the amount of times students went off screen. Student workers set up the lecture on a projector and helped coordinate student interaction with the instructor. Of the five sections of the course, two were taught in this manner: sections 1 and 5. The remaining sections (2, 3, and 4) were taught face to face by a different faculty member. This continued until the faculty member returned about a month into the semester, right before exam one.

This remote mode of instruction brought some significant challenges. Power interruptions, common in the instructor's home country, interrupted Zoom lectures. Some class time each day needed to be devoted to setup. Since this solution was implemented rapidly, there was not enough hardware to supply each classroom with appropriate speakers and a dedicated computer. Student workers needed to ensure that the appropriate hardware traveled to the classrooms and was set up appropriately. Finally, interaction between students and faculty was less fluid than in a traditional face to face setting; microphones and audio don't always work in large groups and monitoring the text chat can be difficult to juggle with the delivery of material.

Before exam one, students were administered a voluntary survey to assess their perceived level of learning achieved in the course along with their prior experience. The full form of the survey can be found at `https://forms.gle/4C9w34zvTwUiAXFP8`. Interesting aggregate responses for the perceived learning level questions will be presented below.

After the first midterm exam, the performance of the sections were compared using ANOVA with the null hypothesis that the scores for the sections were indistinguishable. Individual questions on the midterm and the total score were analyzed to look for patterns in how sections performed.

Exam two was evaluated in the same way to determine if there were any long term effects (positive or negative) from having a remote instructor teaching in class synchronous learning.

3 Measures

Before the first exam, students were administered a voluntary survey to assess their perceived level of learning achieved in the course along with their prior experience with programming and related technologies. Across all sections, 88 students had the remote instructor for the first month of classes, while 141 students had a more traditional face to face experience for a total of 229 students. Of those, only 75 students responded to all of the likert scale questions in the survey; 22 students from Sections 1 and 5 (remote instructor), and 53 students from sections 2, 3, and 4 (face to face). The preparedness questions in the survey had students rank their agreement on a standard 5 point Likert scale (Strongly Disagree, Disagree, Neutral, Agree, Strongly Agree); the range of responses from these questions is shown in Figure 1.

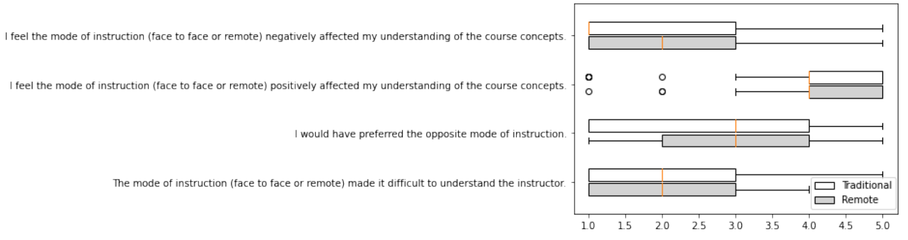


Figure 1: Likert scale preparedness responses; circles represent outliers.

The first thing to note is that for the most part, the survey responses are similar in nature. While the low response rate prevents strong conclusions to be drawn from this data, there are some trends that are of interest. In the rest of this discussion, the disagree and strongly disagree responses will be aggregated and presented as a percentage of the total. Similarly, the agree and strongly responses will be aggregated.

3.1 Did students believe the mode of instruction negatively affected performance?

This was addressed by the first question in the survey (Figure 2). There is a small shift in the responses with an increase from 9 percent in the tradition mode to 19 percent in the remote mode agreeing that their performance was negatively affected. There is a corresponding decrease in the percentages for the belief that scores were not negatively affected from 73 to 63 percent.

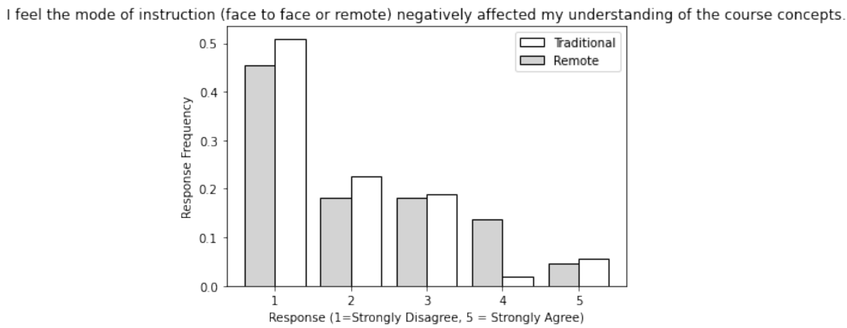


Figure 2: Percent Responses: Question 1

3.2 Did students believe the mode of instruction positively affected performance?

The second question shifted the viewpoint slightly and asked directly if students felt the instruction mode positively affected their performance. One would expect comparable results to the first question. But in this case, the difference in the percentages was reduced (Figure 3). In both groups, about 80 percent of students felt their performance was positively affected. The percentage believing the opposite rose from 9 percent for traditional group to 14 percent in the remote group. In both cases, we see a reduced difference in the perceptions between the two groups.

3.3 Did students prefer the other instruction mode?

The third question asked for their preference (Figure 4). For both groups, about 1/3 of all respondents indicated a preference for the other mode of instruction. In the traditional group, about 1/2 of the respondents preferred to stay with it, while only 1/3 of the respondents in the remote group prefered remain remote. It is interesting that there was a significant preference for switching in both groups.

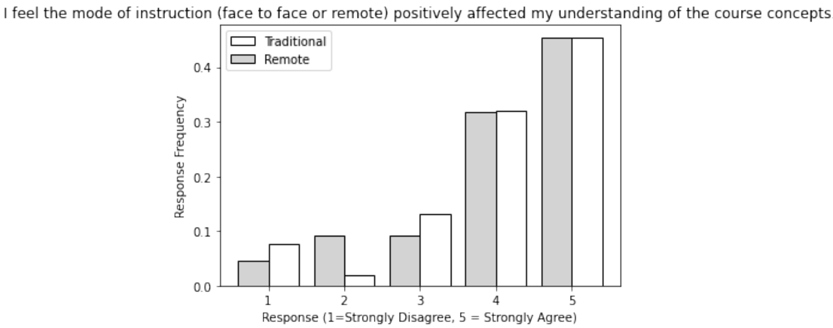


Figure 3: Percent Responses: Question 2

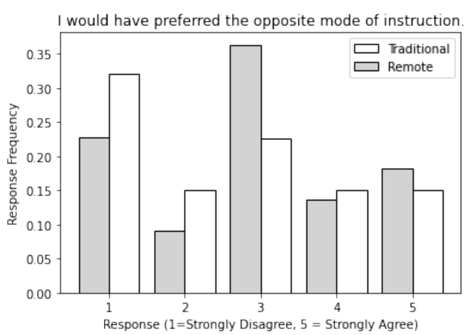


Figure 4: Percent Responses: Question 3

3.4 Was the instructor hard to understand?

In the fourth question (Figure 5), clarity of presentation for the two modes is explored. Both modes are close in percentages with a slight advantage for the traditional mode. About 13 percent felt their mode of delivery was hard to understand and about 70 percent did not.

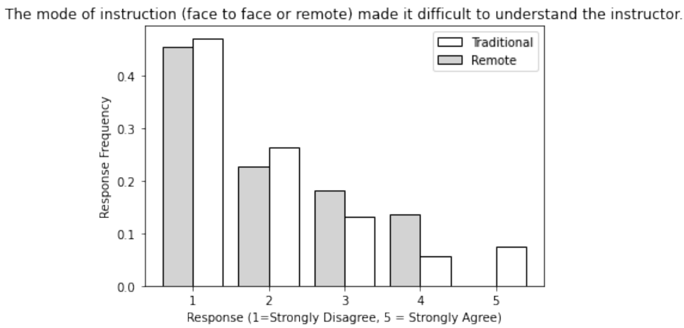


Figure 5: Percent Responses: Question 4

Overall, the perceptions across the two groups were remarkably similar, with a slight advantage held by the traditional group. Notably, more of the remote group felt that their mode of delivery affected their understanding of the material and about half of the traditional group did not have a preference for the remote mode.

4 Effects at 4 Weeks

Short term effects of the alternate instruction mode were measured by evaluating individual questions and overall performance on the first Midterm Exam which took place roughly four weeks into the semester, shortly after the remote instructor returned. Testing conditions were as follows:

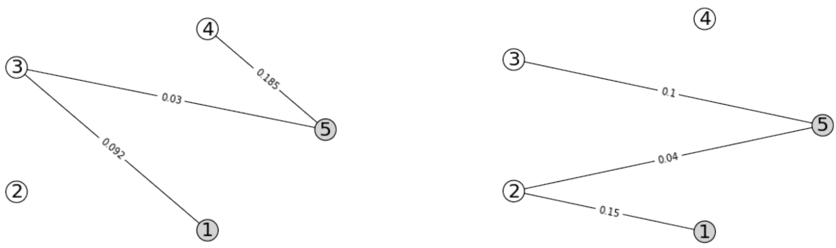
- Students were allowed a single hand written letter paper cheat sheet (single sided) that was collected at the end of the exam.
- Exams were administered in two cohorts in one evening; sections 1,2, and 5 took the exam in the first 50 minutes, followed shortly by sections 3 and 4.
- Each cohort had two different but comparable versions of the exam to make it more difficult to get answers from others.

- The exam consisted of four questions each worth 10 points. Question 4 was comprised of 5 multiple choice questions pulled from a bank.

Table 1 shows per question and overall averages of scores on Exam 2. Figures 6 and 7 shows which pairs of sections have statistically significant differences as reported by ANOVA. An 80% confidence level was chosen to be more sensitive for possible differences with a moderate number of samples in each class group and to account for the number of uncontrollable variables.

Table 1: Section means per question and overall; questions worth 10 points each

	Q1	**Q2**	**Q3**	**Q4**	**Total**
Section 1	7.34	8.79	6.66	8.59	78.97%
Section 2	7.68	8.86	6.91	8.98	81.34%
Section 3	7.79	8.98	7.24	8.88	82.58%
Section 4	7.61	8.95	7.04	8.63	81.01%
Section 5	7.43	8.82	6.60	8.40	78.67%



(a) Question 3 (b) Question 4

Figure 6: Midterm 1 by question. Connected nodes are distinguishable with 80% confidence. Edge weights represent ANOVA P values. Shaded nodes represent sections with remote instructor.

All sections were indistinguishable for questions one and two with a significant difference at 80% confidence and are not shown. Looking at the rank order of sections on question 3 from lowest to highest on question 3 we have: section 5, section 1, section 2, section 4, section 3. The two remote sections (5 and 1) came in behind the three face to face sections. The highest performing section (face to face) was distinguishable from both of the remote sections, and the lowest performing section (remote) was distinguishable from two of the face to face sections.

When we consider question 4 (multiple choice) the order is: 5, 1, 4, 3, 2. Again, both of the remote sections were behind the face to face sections. We also see the same pattern where the highest and lowest performing sections are distinguishable.

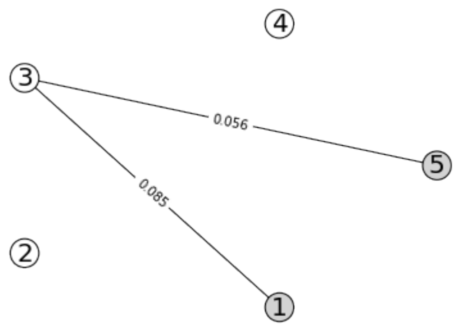


Figure 7: Midterm 1 overall. Connected nodes are distinguishable with 80% confidence. Edge weights represent ANOVA P values. Shaded nodes represent sections with remote instructor.

For the total percentages, the order is 5, 1, 4, 2, 3, but the only significant difference is between the lowest (remote) and highest (face to face) sections. This kind of distribution could be explained by differences in the students assigned to the various sections.

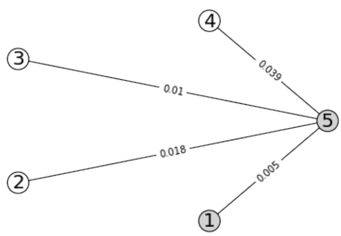
5 Effects at 8 Weeks

Longer term effects of the alternate instruction mode were measured by evaluating individual questions and overall performance on the second Midterm Exam which took place roughly eight weeks into the semester. Exam conditions were identical to those of Exam 1. Table 2 shows per question and overall averages of scores on Exam 2. Figures 8 and 9 show which sections have statistically significant differences as reported by ANOVA.

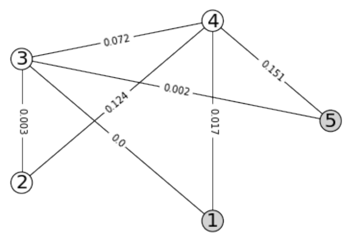
For the second exam, all of the sections were indistinguishable on question 4 (multiple choice). Again, the rank order of the sections is presented for each question: Question 1 is 5, 4, 2, 3, 1; Question 2 is 1, 5, 2, 4, 3; and Question 3 is 5, 4, 2, 1, 3. Compared to the first exam, the lowest performer (section 5 remote) and the best performer (section 3 face to face) maintain

Table 2: Section means per question and overall; questions worth 10 points each

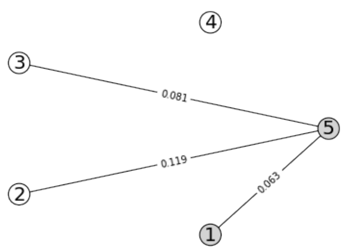
	Q1	Q2	Q3	Q4	Total
Section 1	7.9	6.48	5.98	9.1	72.92%
Section 2	7.74	6.79	5.86	9.19	73.26%
Section 3	7.85	7.94	6.03	9.29	76.72%
Section 4	7.64	7.36	5.45	9.09	72.72%
Section 5	7.01	6.91	5.12	9.12	70.16%



(a) Question 1



(b) Question 2



(c) Question 3

Figure 8: Midterm 2 by question. Connected nodes are distinguishable with 80% confidence. Edge weights represent ANOVA P values. Shaded nodes represent sections with remote instructor.

their respective rankings and show a statistically significant difference. With the remaining sections, the rankings are less stable with section 1 (the other remote) exhibiting performance over the entire range of best to worst.

For the percentages on the total grade for the exam, the same pattern emerges: The two extremes section 5 (remote) and section 3 (face to face) exhibit significant differences. The remaining sections cluster together and are indistinguishable from each other.

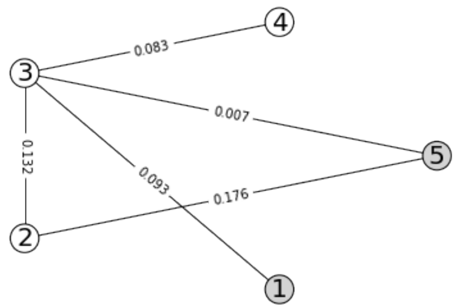


Figure 9: Midterm 2 overall. Connected nodes are distinguishable with 80% confidence. Edge weights represent ANOVA P values. Shaded nodes represent sections with remote instructor.

The overall grades of the students after the second exam was graded give a more wholistic view of student performance, shown in Table 3 and Figure 10. There is a significant difference between section 3 (face to face) and all the other sections. This supports a conclusion that any difference due instructor mode has been washed out and the difference with section 3 is likely due to some other factor.

Table 3: Mean and Standard Deviation after Exam 2

Section	**Mean**	**Std. Dev.**
1	72.92	8.77
2	73.26	10.49
3	76.72	11.48
4	72.72	10.35
5	70.16	11.4

6 Challenges to Validity

Looking at the data, the face to face sections had higher averages than the remote sections. While there may have been a difference in the early (Midterm 1) outcomes, it was small. Determining the cause of difference is complicated

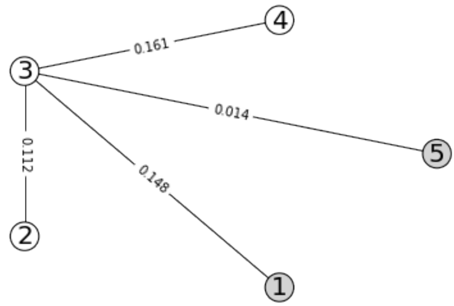


Figure 10: Overall course average after Exam 2. Connected nodes are distinguishable with 80% confidence. Edge weights represent ANOVA P values. Shaded nodes represent sections with remote instructor.

as there are other possibilities besides the instruction mode. Unfortunately, these other possibilities could not be directly controlled for in this study.

6.1 Instructor

Of necessity, the in-person and remote mode sections were taught by two different instructors. The instructor for the in-person sections has more years of teaching experience and was the primary developer of the current course content. In addition, this was the first time that the remote instructor was teaching the course. The extra familiarity with the material and knowledge of exam structure ahead of time may have aided the face to face instructor, however subconsciously. This could be mitigated to some extent by there being extensive use of shared materials in the forms of slides, quizes, and assignments.

The traditional instructor still insists on masks in the classroom, which may explain somewhat the answers to the survey question about understanding the instructor. This may, however, offset any difficulties introduced by the introduction of technology.

6.2 Students

Students create several additional variables in this study; while predominantly a graduate level class taken first in the sequence, some sections have more undergraduate students than the others. Additionally there is no way to control for the way students were distributed into the different sections. It is possible that one section contains "better" students, or students with more experience or problem solving acumen. This would be supported by the fact that when sections were ranked by average score on the midterm, the order was the same except for a swap in the third and fourth.

6.3 Exam timing

Finally, due to the size of the sections, exams were not taken by all students simultaneously. The exam was given in two back to back 50 minute time slots. Sections 1, 2, and 5 took the exam first, followed shortly by sections 3 and 4. This does present the opportunity for some information leakage between exam cohorts. The allowance of a single sided "cheat sheet" with the student's name that got turned in with the exam was an attempt to mitigate and discourage information leakage, but it is difficult or impossible to measure the exact effect on exam performance. On a per question basis, only the second question of the second exam showed a significant difference between sections 3 and 4 with the earlier sections.

7 Conclusions

While not ideal, having a course where the instructor is remote and lectures over video to a group of students that are viewing from a single location can be effective. Our experience showed that while the students may have perceived a difference, their performance mostly did not show a significant difference. Further, when there was a difference, it was small and could have been attributed to the random distribution of students into sections or exam timings. We believe that substantive shared lecture notes, assignments and teaching assistants helped to promote a uniform experience for the students. Further, keeping the remote students in the classroom at a regularly scheduled time, contributed to better participation.

References

[1] Wahab Ali. Online and remote learning in higher education institutes: A necessity in light of covid-19 pandemic. *Higher education studies*, 10(3):16–25, 2020.

[2] Naomi T Fitter, Nisha Raghunath, Elizabeth Cha, Christopher A Sanchez, Leila Takayama, and Maja J Matarić. Are we there yet? comparing remote learning technologies in the university classroom. *IEEE Robotics and Automation Letters*, 5(2):2706–2713, 2020.

[3] Tawnell D Hobbs and Lee Hawkins. The results are in for remote learning: It didn't work. *The Wall Street Journal*, 2020.

[4] Hani Morgan. Best practices for implementing remote learning during a pandemic. *The clearing house: A journal of educational strategies, issues and ideas*, 93(3):135–141, 2020.

[5] Suzanne Young and Mary Alice Bruce. Classroom community and student engagement in online courses. *Journal of Online Learning and Teaching*, 7(2):219–230, 2011.

Learning Analytics Finds That a Shared Course May Improve Technology Students Retention*

Chunbo Chu and Jiang Li
Computing Sciences and Mathematics
Franklin University
Columbus, OH 43215
{chunbo.chu,jiang.li2}@franklin.edu

Abstract

We investigated students' performance data in a Computer Science program by applying Learning Analytics to gain a better understanding of the retention issue. We found that students who have taken a soft introduction course in information systems shared by multiple technology majors tend to transfer to non-CS technology majors instead of transferring to non-technology majors or dropping out. Our research suggests that a holistic strategy on curricula design may improve retention.

1 Introduction

Retention is a major concern for many higher education institutes around the world. Retention data is important, not only because significant money is lost to non-returning students, but also because such data provides a measurable metric for better understanding why students leave before graduating [7]. To make informed decisions, academic leadership must have a good understanding of what their students are struggling with and why.

Learning Analytics is the measurement, collection, analysis, and reporting of data about learners and their contexts, for the purposes of understanding and optimizing learning and the environments in which it occurs [1]. It is a powerful tool that can help us gain insights into the retention issue for better decision making. There are many methods applicable to Learning Analytics. The most popular are related to prediction, clustering, and relationship mining.

In this study, we investigated the students who started as undergraduate Computer Science majors in a private, non-profit, teaching-intensive university in the Midwest U.S.A. over a 10-year period. This open-admissions university specializes in working adult education delivered through e-learning. It offers five undergraduate majors in technology: Computer Science, Cybersecurity, Information Systems, Information Technology, and Web Development. While the aggregated enrollment of these majors has been growing steadily, anecdotal evidence has been observed that many students did not complete their programs, especially Computer Science. We used relationship mining to study the relationships among variables influencing Computer Science students' retention. Our focus is on the change of students' status and their academic performance in the last trimester before they left the program.

2 Related Work

There are persistent challenges in producing and retaining STEM talent in the United States to meet the current workforce demands, as documented in several national reports. Although about 28% of all U.S. college students select a STEM major, more than half switch to a non-STEM field or leave postsecondary education without earning any credentials [4].

One third of undergraduate students enrolled in 2011 were age 25 years or older, often balancing their academic studies with work and family obligations [8]. The population of nontraditional students is projected to increase significantly. Such students may face even more challenges in completing their degrees.

Around the world, e-learning (online learning) has become very popular, especially during the COVID-19 pandemic. Compared to traditional classroom learning, e-learning has many benefits, such as higher flexibility, development of technical skills, continuous evaluation, and individual and collaborative activities [5]. Generally, e-learning makes education more accessible and affordable. However, e-learning courses also result in higher dropout rates because distance education may create a sense of isolation in students who can feel disconnected from the other students, the instructors, and the university [3].

3 Research Methodology

There were 1,302 students enrolled in the Computer Science program in their first trimesters at the university under investigation between the Fall 2011 and

Summer 2022 trimesters. A total of 14,897 data points were collected. Each data point contains: student ID, course number, trimester, grade, and program code. It represents a student and a course he or she took in a given trimester. Out of these data points, 86% of the courses were taken online (12,856), and 14% were taken face to face (2,041). We did our analysis in R [11], and produced figures by using the `ggplot2` package [9]. The standard deviations of the course proportion before the transfer were estimated using the bootstrap samples.

4 Findings

For the purpose of retention consideration, we defined a student's status during their lifetime at the university as follows:

- Active: student did not graduate or transfer (defined below) and has taken at least one course in the most recent three trimesters.
- Transferred: student changed his or her program code from "CS" to another major at the university.
- Graduated: student has taken the CS capstone course.
- Inactive: student did not graduate or transfer and has not taken any course in the most recent three trimesters.

Figure 1 illustrates how a student's status may change over time.

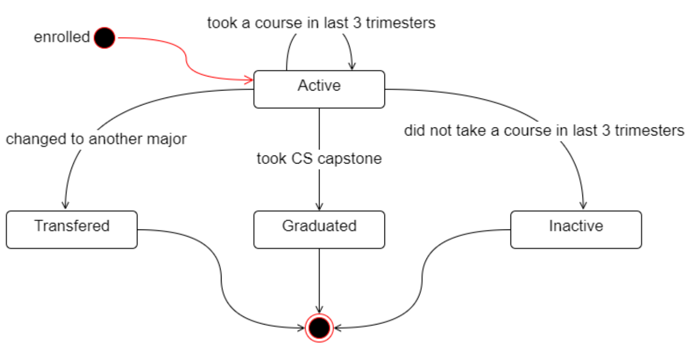


Figure 1: Students Flow

For a long time, we have observed that many students were initially attracted to the Computer Science major but graduated from other non-CS tech majors: Cybersecurity, Information Systems, Information Technology, and Web Development. This assumption was confirmed by Figure 2. During the time span of the data collected, 10% of the CS majors transferred to other non-CS tech majors, and 12% transferred to non-tech majors. More than half of the students went inactive. We do not have data to show whether they transferred to other institutions, dropped out of college, or just needed a long break before returning to the program.

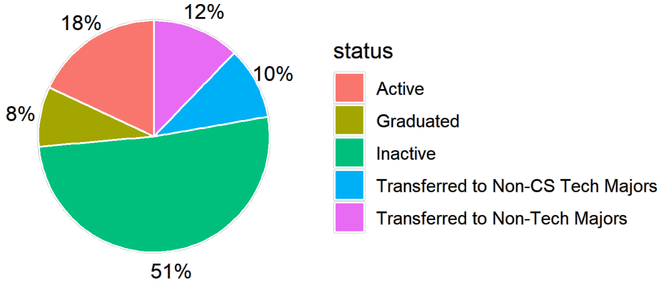


Figure 2: Students Retention

Other studies show that college dropout rates average 40% for undergraduate students in the U.S. [10]. The leading reasons are financial concerns, academic disqualification, and difficulty in balancing life and college [2]. Though the inactive status is not the same as dropout, it can be used as an approximation. Considering that the university offers open admissions and primarily attracts working adults, factors such as academic disqualification and difficulty in balancing work, life, and college may have an even larger impact here than at a traditional college. It is reasonable to assume that the 51% attrition rate of the CS major is on par with the nationwide dropout rate. Figure 2 puts the Computer Science retention issue in the right context and sets reasonable expectations on potential remedial measurements.

We split the students who left the Computer Science major (identified as Transferred or Inactive) into the following three groups:

A. transferring to non-CS technology majors

B. transferring to non-technology majors

C. becoming inactive

We identified the top five courses frequently taken by the three groups respectively in the last trimester before changing status. See Figure 3. Clearly, all these students struggled with mathematics and programming. This is especially true for CS1 and CS2[1] (appearing in all three groups), Discrete Math, and College Algebra (appearing in two groups).

[1] Though there is no wide agreement on what occurs in CS1 and CS2 courses, we use the

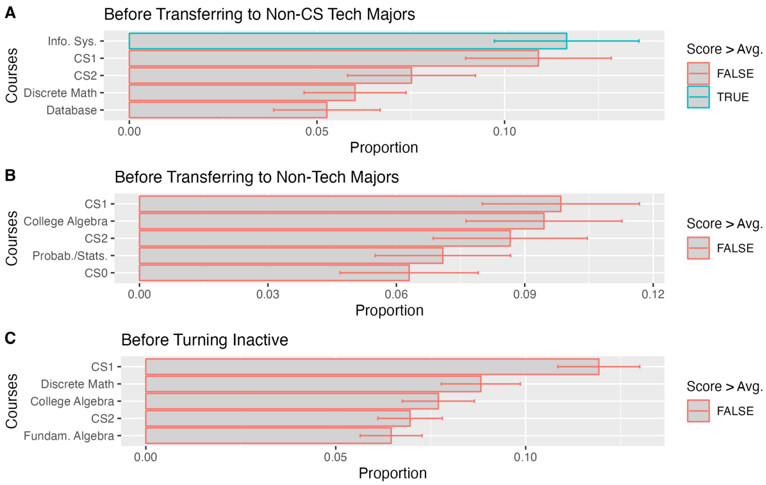


Figure 3: Courses Before Status Change. Proportion of a course is the percentage of students belonging to a group out of all students taking that course. Error bar represents the standard deviation.

CS majors would take almost all these courses in their first year. Anecdotes from instructors and student advising suggest that many students who have struggled in these fundamental courses dropped the classes and did not return to the program.

Sub-figure A, however, tells us a very interesting story that we did not expect. Just like the other two groups, Group A did poorly in the fundamental math and programming courses, so they stopped pursuing the Computer Science degree in the following trimester. But instead of completely abandoning technology majors or college altogether, they transferred to non-CS technology majors. They would be counted towards the attrition of the Computer Science major but not toward the attrition at the department level.

Compared to Sub-figures B and C, A includes a course in which the students actually did well: their average score in the Information Systems Architecture course was higher than the average score of all students taking the class. This is a non-technical course offering a conceptual survey of general information systems from the business perspective. Topics include computer hardware and software, database and big data, network and cloud computing, business intelligence and analytics, information systems security, and ethical and social

terms to refer to the first two introductory courses of object-oriented programming in Java. They cover the most commonly taught topics in many institutes [6].

issues. It is a major area elective course in Computer Science and major area required course in Information Technology, Information Systems, and Cybersecurity. It does not require any math or programming classes as prerequisite. Therefore, this course is less challenging than other courses in the list such as CS1, CS2, and Discrete Math.

Could this course hold the key? We think the following may be a plausible explanation.

According to interviews with student advising, many students who claimed to be Computer Science majors to begin with did not really understand what Computer Science is about or what it takes to be successful. They were just attracted to the most widely recognized brand name to study computers or allured by the prospect of getting a better job with a CS degree. When they struggled in the fundamental math and programming courses, their interest and confidence diminished, which led to the subsequent dropout from Computer Science. Technically, these courses filtered unprepared students who would have been less likely to be admitted in the Computer Science major at selective admissions universities.

However, the non-technical nature and the variety of the topics in the Information Systems Architecture course open a new window on computing. Taking the course helped the challenged students maintain an interest in the broader computing discipline. Confidence in continuing in the technology majors was built from a different perspective. This course engages these students and bridges them to the non-CS technology majors that better suit their aptitude and interest. It also makes financial sense since the credit for this course can still be used towards graduation.

Based on this hypothesis that reveals the symbiotic relationships among different technology majors, we think that just offering multiple majors in computing technology may not be enough to retain students in the department. At the curricular level, it is important to have one or more courses that "gently" cultivate students' interest in the general discipline of computing in early stages. These courses should be shared by some, if not all, related programs during the first year. They should expose students who are unprepared but truly interested in computing to the many different aspects of computing technology so that they can re-evaluate whether their initial major of choice, say Computer Science, is really what they want or can handle. This approach would enable students to make informed decisions to pursue the major that suits them the best as early as possible. As a result, student stickiness is increased, which improves retention at the department level.

Our finding shows that related academic programs should be designed with early opportunities to help students build confidence and learn more about their initial and alternative majors. This would give them better chances to be successful. We suggest that the academic leadership employ a holistic strategy for addressing the retention issue. Reducing the student attrition rate does not necessarily mean that every program must have the same target percentage point. It means that students should find it easier to identify and transfer to (or stay in) the major that best suits their real interests and aptitudes, which is the key to retention.

5 Conclusion and Future Work

In this research, we investigated the students' performance data in a Computer Science program over the stretch of 10 years. We applied Learning Analytics approaches to get a better understanding of the retention issue. We found that, among students who dropped out of the Computer Science major, those who took a soft introduction course in information systems architecture that is shared by multiple technology majors are more likely to transfer to non-CS technology majors than transferring to non-technology majors or dropping out of the university. This finding suggests that related programs are not isolated. They support and draw support from one another like living organisms in an ecosystem. A holistic strategy on the retention issue and curricula design is warranted.

Student performance data only tells a partial story. We would like to continue this research by collecting more data, such as previous experience in computing, gender, age, race etc. A follow-up survey will get direct responses from the students about why they did not return for Computer Science. We also plan to build and train a predicative model to support early intervention on students in jeopardy of dropping out.

References

[1] Call for papers of the first international conference on learning analytics and knowledge (lak). `http://www.wikicfp.com/cfp/servlet/event.showcfp?eventid=11606`.

[2] Imed Bouchrika. College dropout rates: 2022 statistics by race, gender & income. `https://research.com/universities-colleges/college-dropout-rates`.

[3] Laura Calvet and Angel Juan. Educational data mining and learning analytics: differences, similarities, and time evolution. *RUSC. Universities and Knowledge Society Journal*, 12:98, 07 2015.

[4] Xianglei Chen. Students who study science, technology, engineering, and mathematics (stem) in postsecondary education. stats in brief. nces 2009-161. *National Center for Education Statistics*, 2009.

[5] Thanasis Daradoumis, Angel A. Juan, Fernando Lera-López, and Javier Faulin. Using collaboration strategies to support the monitoring of online collaborative learning activity. In Miltiadis D. Lytras, Patricia Ordonez De Pablos, David Avison, Janice Sipior, Qun Jin, Walter Leal, Lorna Uden, Michael Thomas, Sara Cervai, and David Horner, editors, *Technology Enhanced Learning. Quality of Teaching and Educational Reform*, pages 271–277, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[6] Matthew Hertz. What do 'cs1' and 'cs2' mean? investigating differences in the early courses. In *Proceedings of the 41st ACM Technical Symposium on Computer Science Education*, SIGCSE '10, page 199–203, New York, NY, USA, 2010. Association for Computing Machinery.

[7] Watermark Insights. Why is college retention important? `https://www.watermarkinsights.com/resources/blog/retention-vs-enrollment`.

[8] Gail Markle. Factors influencing persistence among nontraditional university students. *Adult Education Quarterly*, 65(3):267–285, 2015.

[9] R Core Team. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria, 2017.

[10] ThinkImpact. College dropout rates. `https://www.thinkimpact.com/college-dropout-rates/`.

[11] Hadley Wickham. *ggplot2: Elegant Graphics for Data Analysis.* Springer-Verlag New York, 2009.

Implementation and Evaluation of a Virtual Hackathon in an Urban HSI Community College During COVID-19*

Mohammad Q. Azhar[1],
Ada Haynes[2], *Marisa Day*[3] *and Elizabeth Wissinger*[4]
[1]*Computer and Information Systems*
Borough of Manhattan Community College, CUNY
New York, NY 10007
mazhar@bmcc.cuny.edu
[2]*Sociology and Political Science*
Tennessee Tech University
Cookeville, TN 38505
[3]*Research Foundation Of The City University Of New York (Inc)*
New York, NY 10036
[4]*Social Sciences, Human Services, and Criminal Justice*
Borough of Manhattan Community College, CUNY
New York, NY 10007

Abstract

This paper shares the analysis of our quantitative findings regarding the impact of a virtual informal collaborative experiential learning activity on diverse students' computational thinking, critical thinking, and self-efficacy in STEM activities. Designed as part of an ongoing National Science Foundation sponsored project to provide underrepresented minority (URM) students from underserved economic backgrounds with real-world career preparation and technical education across disciplines through collaborative project activities using cutting-edge technologies, the Hackathon for Social Good was implemented during the COVID-19 shutdowns in a New York City community college in lower Manhattan. Students worked in teams to innovate practical solutions to global problems with mentor support from both academia and the tech industry. This intervention drew 36 students from Computer Science, Business, and Sociology classes, who worked with volunteers and alumni during a full-day event in the Fall of 2021, using AI and data science to design culturally sensitive data-driven solutions for real-world problems. The tracks covered the following topics: Zero Hunger, Clean Water, and Sanitation, Green Consumption, Racial Justice, Quality Education, Good Health, and Well Being. The two main objectives of this project are as follows: (1) Design a remote interdisciplinary one-day experiential collaborative learning environment to engage URM teams of students from a community college in applying computational thinking to develop solutions for social good. (2) Conduct research on our intervention to study its effect on students' self-efficacy, as well as their knowledge of, and comfort with, computational thinking, critical thinking, problem-solving, and STEM. The evidence gathered from qualitative and quantitative data indicates that using these mechanisms to infuse CT into student learning across disciplines has several positive outcomes. Students reported increased leadership skills, comfort with teamwork, problem-solving, and critical thinking. A quantitative study specifically showed a positive impact on student confidence in their ability to do CT and improved their sense of efficacy in impacting the world outside of the hackathon.

1 Introduction

Hackathons have been studied as a site for collaborative problem solving using technology to create solutions to real-world problems in one- to two-day challenges [10]. Incorporating project-based learning activities to engage students in STEM in an informal environment has been demonstrated to increase STEM engagement and positively expose students to a variety of STEM career options [14, 12]. Hackathons are also increasingly understood as a valuable educational tool [16]. A gap remains in the extant research on designing virtual experiential learning environments, such as virtual hackathons, for community college students. To address these issues, we designed, implemented, and evaluated our virtual hackathon during Fall 2021 at an urban Hispanic Serving Institution (HSI) community college for URM students. We included students from Computer Science, Business, and Sociology courses to cultivate interdisciplinary participation. Thirty-six students joined volunteers, business leaders, and alumni in a full-day event, including speeches, question-and-answer sessions with industry leaders, hack time, judging, and awards. The virtual hackathon activities had a discernible impact on URM students' computational thinking, critical thinking, creativity, and self-efficacy.

2 Related Work

Hackathons share common traits of collaborative problem-solving and technological innovation. Virtual hackathons have a unique position in contrast to a physical hackathon, in their ability to facilitate collaboration from individuals from a variety of locations, across gender and racial diversity. Flexible participation in virtual hackathons generates innovative ideas from a wide range of participants using free, widely distributed digital resources [16]. For example, the online collaboration provided by the "COVID-19 Flatten The Curve Hack #flattenthecurvehack," during the COVID-19 pandemic, incorporated international participation from 2000 individuals who worked collaboratively online to innovate solutions to the challenges of COVID-19 [20]. University-sponsored hackathons provide students with hands-on opportunities to develop new technical skills, connect with industry mentors, and work in a team with peers to solve real-world problems [8, 10, 13, 16, 23]. Virtual university-sponsored hackathons, similar to our intervention, facilitate university-industry collaboration, particularly during times such as the COVID-19 pandemic when colocation was not possible [8] and are increasingly understood as valuable collaborative instruments for problem-solving [4, 18].

Hackathons, by their design, are a hotbed for informal project-based learning (PBL) [9]. The effectiveness of PBL has been documented in studies on its effect on the choice of major, career aspirations, and overall student attitudes, particularly for URM students [3], and offers a site for culturally responsive pedagogy to thrive [5]. Prior research suggests that inquiry-based hands-on scaffolded learning, such as the kind we implemented in our project, can serve as a critical component in combating inequalities in computing for URM students, connecting computing to society, and using scaffolding to train students to apply abstractions and models in collaborative projects [9].

PBL provides hands-on opportunities for students to exercise their unique strengths, assets, and agency, in contrast to the deficits-based approach which is often found in interventions targeting URM in STEM [11]. Incorporating scaffolding into project-based learning allows students to build familiarity with concepts in complex domains and reduces the cognitive load in the learning process [7]. We followed this scaffolding model by incorporating skill-building preparatory materials and team building. Professional skills are one of the strongest sites of positive change for students who participate in hackathons [18]. Hackathons replicate the problem-solving and collaboration required in the business world and often produce portfolio projects by participants upon completion, which can be used to demonstrate students' capabilities to future employers. Digital badging has been studied to show real-world benefits in career development, by providing third-party credentials, to display on a website, Linked-In, or resume, demonstrating competence and skill in a technical area [6]. We incorporated these findings in our project design, by including professional skill development; providing students with industry mentorship and networking; and the opportunity to create portfolio projects, along with digital badging.

Community-engaged engineering helps students develop a design for justice lens which embeds socio-technical thinking skills into the learning process [10]. The subject matter of our challenge engaged students in solving global problems to incorporate these benefits into the student experience [12, 17]. The ideation and design of this project carefully build on the successes outlined in the literature by incorporating these concepts in an intervention developed for the unique urban URM community college context.

3 Designing a Virtual Hackathon Co-Curricular Learning Environment

Our virtual hackathon engaged interdisciplinary student teams remotely using Zoom for one full Saturday. The collaborative, synchronous activity was supported by faculty and student peer mentors and judged by industry professionals. Hackathon projects for social good were designed to be scaffolded, and culturally responsive, to encourage student engagement in solving real-world problems [19]. Student teams collaboratively used cutting-edge technology to develop innovative solutions to challenging problems which incorporated a global Call for Code challenge [1] and the UN's sustainable development goals (SDGs) [2]. Call for Code is a global initiative led by IBM to apply crowdsourced coding solutions to societal issues, requiring students to collaborate virtually to design a new or speculative product to solve real-world problems. Teams designed projects for six tracks of UN SDGs: Zero Hunger, Clean Water, and Sanitation, Green Consumption, Racial Justice, Quality Education, Good Health, and well-being. Solutions incorporated AI/data science and were communicated by a website and video.

Faculty in Computer Science, Sociology, and Business departments collaborated in the lead-up to Fall 2021 to facilitate the participation of an interdisciplinary cohort of students. Students were encouraged to participate through announcements made in all classes including a video featuring student testimonials from previous hackathons at our institution. In the weeks leading up to the hackathon, students had the opportunity to meet across disciplines on Zoom in the Success and Innovation Lab, an ongoing virtual site active during the semester for student education and innovation. In addition to becoming familiar with technical topics and onboarding, facilitated by the Computer Science Faculty and the BMCC Computer Programming Club, students self-selected their teammates in these meetings. Some teams expanded to include members who registered on the day of the event, which was added by hackathon administrators. Peer mentors led morning workshops on three tracks of technical skills that students could self-select to join.

Throughout the hack period, student teams were separated into Zoom rooms with their teams to collaborate, and receive mentoring and insight from industry experts and peer mentors. At the end of the day, team presentations were judged by industry experts who chose six winning teams. While judges deliberated, students participated in a focus group about their experiences. Student projects became portfolio pieces for student career development, housed on the DevPost "Home for Hackathons" site.

4 Research Methods

Our research employed quantitative survey research and qualitative focus groups. The survey research was conducted at the end of the Hackathon day using Qualtrics software. We administered a single survey developed by the research team to gather demographic data and ask questions about feelings and attitudes prior to the hackathon, and post-hackathon to capture shifts in student attitude attributable to engagement with the hackathon. This survey also provided the benefit of having pre and post-responses as matched pairs for almost all questions and students. The survey explored the impact of an informal, virtual, experiential learning activity, e.g. the hackathon, on students' knowledge of, and comfort with, computational thinking, critical thinking, problem-solving, and self-efficacy with regard to STEM activities.

4.1 Human Subjects

All human subjects' guidelines were followed in this study, including submission and approval as exempt research from the university's Institutional Review Board (IRB).

4.2 Survey

Survey questions were modified from surveys by [15]. Questions included demographic information. Students also completed 5-6 point Likert scale questions related to their STEM knowledge (5-point), comfort (5-point), and self-efficacy (6-point) prior to and post the hackathon. The term "computational thinking" [21, 22] was adapted from these sources and defined for students throughout the hackathon and in the survey as "thinking logically to solve problems and abstracting principles and applying them in other situations."

5 Quantitative Findings

Thirty-four students are included in the sample, as within the group of 36 students who completed the survey, 2 were under 18. The majority of the students were in a computer science-related major. Forty-four percent of participants selected that neither of their parents attended college. Twenty-five percent reported that one or both of their parents completed a university degree. Fifty-three percent of the students work on or off campus, with 26% working full-time off campus. Of the respondents, 62.5% received financial aid, and 37.5% did not receive financial aid. While 88% of the respondents selected that they chose their major because of interest in the subject matter, 56% of students chose their major because of potential pay. Another 47% of students selected the prospect of making a difference and an equal percentage selected work conditions and expected benefits as their rationale for choosing their major.

The project did an excellent job of recruiting female students to the hackathon with almost half (41%) identifying as women, with the remaining 59% of the participants identifying as men. This number is much higher than the percentage of women students in computer science nationwide. Men were more likely than women to have 'always thought that they would study in this field' and indicated their interest in 'the social aspects of jobs.' Men's interests were more influenced by faculty members and other students, while women were more influenced by parents. There was a broad distribution of race and ethnicity among the participating students with 19% of the students self-classified as Latino(a), 3% preferred not to answer. Only 18% of students classified themselves as white.

5.1 Paired t-tests for Knowledge

Students rated their knowledge of computational thinking, problem-solving, critical thinking, and Science, Engineering, Technology, and Mathematics (STEM) on a five-point Likert Scale from "not knowledgeable at all" to "very knowledgeable" prior to and post the hackathon, compared using paired t-tests. On average, students' knowledge of computational thinking was lower prior to the hackathon than after the hackathon. This improvement was statistically significant as in Figure 1. Students' knowledge of problem-solving was significantly lower prior to than post the hackathon. Students' self-reported knowledge of critical thinking was significantly lower prior to the hackathon. Students reported knowledge of STEM was also significantly lower prior to the hackathon (Figure 1).

	Pre-survey		Post-Survey		
	Mean	SD	Mean	SD	t
Computational Thinking	3.31	1.03	3.63	1.008	-3.754**
Problem Solving	3.66	0.971	4.03	0.861	-4.313***
Critical Thinking	3.63	0.942	3.91	0.963	-3.483**
STEM	3.47	1.016	3.69	0.965	-2.521*
*p<.05 , **p<.01, ***p<.001					

Figure 1: Results of Paired t-tests for Knowledge

5.2 Paired t-tests for Comfort

Students rated their comfort level with computational thinking, problem-solving, critical thinking, and Science, Engineering, Technology, and Mathematics (STEM) on a five-point Likert Scale from “not comfortable at all” to “very comfortable” prior to and post hackathon. The mean level of comfort with each of these items prior to the hackathon was significantly lower than post participation (Figure 2).

	Pre-survey		Post-Survey		
	Mean	SD	Mean	SD	t
Computational Thinking	3.28	0.958	3.72	0.958	-3.999***
Problem Solving	3.53	0.915	4.03	0.897	-4.546***
Critical Thinking	3.68	0.945	4	0.894	-2.997**
STEM	3.47	1.016	3.75	0.916	-3.483**
*p<.05 , **p<.01, ***p<.001					

Figure 2: Results of Paired t-tests for Comfort

5.3 Paired t-tests for Computational Thinking

Students were asked a series of questions in regard to computational thinking based upon a 6-point Likert scale from strongly disagree to strongly agree. Students reported a significantly higher level of agreement that they could apply knowledge of computational thinking to solve problems after the hackathon than prior. Students’ comfort level learning computational thinking concepts significantly increased. Students reported significant gains in their agreement that they could use computational thinking in their daily life. Students were significantly more likely to report that they found computational thinking not boring post the hackathon. The change in the statement "the challenge of solving problems using computational thinking appeals to me" was moderately significant with students reporting higher agreement post the hackathon than prior to the hackathon. Students reported that they would more likely choose to take computational thinking classes if given the opportunity after the hackathon than prior to the hackathon (Figure 3).

	Pre-survey		Post-Survey		
	Mean	SD	Mean	SD	t
Apply Computational Thinking to Solve Problems	4.53	1.107	4.94	0.759	-2.347*
Comfort Learning Computational Thinking Concepts	4.53	0.983	5.06	0.801	-3.947***
Use Computational Thinking in Daily Life	4.5	0.95	5.06	0.716	-3.788**
Computational Thinking Not Boring	4.65	1.199	5.13	0.763	-3.028**
Challenge of Solving Problems Using Computational Thinking Appeals to Me	4.74	1.032	5.06	0.68	-1.834
Choose to Take Computational Thinking Classes	4.69	0.93	5.03	0.778	-2.267*
*p<.05 , **p<.01, ***p<.001					

Figure 3: Results of Paired t-tests for Computational Thinking

5.4 Paired t-tests for STEM

While most of the students were interested in a STEM career prior to the hackathon, many still found STEM intimidating prior to the hackathon. Students significantly reported that they felt more confident in their ability to solve real-world problems related to STEM after the hackathon than prior to the hackathon. Students were asked to rate their level of agreement on a 6-point Likert scale from strongly disagree to strongly agree in their interest in STEM. Students did not have a significant increase in STEM as a possible career choice. Since the beginning average response was over 5 on a six-point scale, it is possible that there was a ceiling effect on this particular item with students participating in the hackathon already having a strong interest in a STEM career. Students did report a significant decrease in their intimidation in STEM. Confidence in their ability to do computational thinking significantly increased. Students also were statistically significantly more likely to report agreement with the statement that they felt that they could make meaningful changes in the world around them after the hackathon (Figure 4).

Almost 99% of the students enjoyed the experience, learned from the experience, liked being part of a team trying to solve problems, and learned

	Pre-survey		Post-Survey		
	Mean	SD	Mean	SD	t
Interest in STEM as Career Choice	5.25	0.762	5.31	0.644	-0.571
Not find STEM intimidating	3.48	1.387	4.03	1.516	-2.326*
Confident in Ability to Solve Real World Problems Related to STEM	4.16	1.194	4.63	1.264	-2.227*
Make Meaningful Change in the World	4.41	1.241	4.91	1.201	-2.490*
*p<.05 , **p<.01, ***p<.001					

Figure 4: Results of Paired t-tests for STEM

important problem-solving skills that they planned on using beyond the experience. All but one student completed the project. A slight majority of students had the opportunity to work with students outside of their major. Over half of the students responded that they definitely planned on participating next year. Twenty-two students responded to the open-ended question about what they gained from participating, their responses corroborate the survey results, including these notable statements:

- I gained experience in Google Sites, problem-solving, resolving conflicts, working in a team, collaborating, seeing how time management affects us, and creating something from scratch to fruition.

- I was able to do some practical programming. I have only done basic programs in classes.

- I gained comfort to work on tech projects.

- Learn to solve the problem by using technology.

Notably, 59% responded about gaining teamwork skills or making new friends. For example, one student responded that they "gained experience in working as a team as well as how to manage and coordinate tasks/roles." Another student responded that they had gained, "friendship, and I learned how to work as a team and help each other out to solve fun and challenging problems." Similarly, 32% of the respondents mentioned gaining problem-solving skills such as the student that responded that they had gained "the ability to work as a team to tackle a problem through programming." Likewise, 32% of the students reported that they had gained leadership skills from participating in the project. One student reported that they had gained "teamwork knowledge, group leading and learning from doing research on the topic" and another reported that they gained "how to organize the project, be an effective team leader."

6 Suggestions for Future Hackathons

Of the respondents, 21 students answered an open-ended question about what could be done to improve the Hackathon. These responses were coded into categories with 55% of the students wanting the Hackathon to last longer and 15% of the students suggesting that the Hackathon not be changed. Each of the next categories included 10% of the students: (1) increase publicity, emphasize no coding is needed (2) more topics, (3) in-person, and (4) allow more time for student preparation in advance. We have noted a need to diversify the hackathon to include more non-CIS students. The all-day Saturday format was difficult for most of the non-CIS students, particularly students who work on weekends. Assigning a grade to the activity may increase participation. The term *hackathon* had negative connotations for non-CIS and URM students. All student self-selected teams were not interdisciplinary. The project team is currently experimenting and evaluating replacing 'hackathon' with the term *ideathon*, including class projects for a grade, with paired interdisciplinary classes to ensure interdisciplinary student teams, and will publish the results in the future.

7 Conclusion and Future Work

Empowering students to solve real-world challenges reframes engaging URM in STEM from a deficits-based approach, to engaging students as active agents of positive change in the world. Our research was motivated by the idea that incorporating a hackathon for students at a two-year college connects their learning to professional application, and connections with industry mentors offer pathways for further career development. Our project bridges the gap between formal and informal learning and the application of knowledge by developing the virtual hackathon model for URM students across disciplines and collecting evaluation data in order to study the efficacy of the effort on student confidence. Results indicate that virtual hackathons can be valuable co-curricular pedagogies. The findings support previous research that found virtual hackathons can improve student skills in problem-solving and teamwork[8]. Our virtual hackathon produced positive impacts found in other project-based learning experiences such as building skills in problem-solving, and critical thinking. Students reported significant increases from this single-day virtual hackathon in comfort and self-efficacy in computational thinking, critical thinking, problem-solving, and STEM. Open-ended responses revealed the hackathon helped many of the students gain leadership skills. Further research, incorporating students with a wider range of interest in STEM, is necessary to understand the impact of hackathons on students' interest in STEM.

We are working to broaden participation in our hackathon and will publish these findings in the future. The future work will include analyses of qualitative data, retention data, and findings on the impact of redesigning the virtual hackathon to engage more students across disciplines.

Acknowledgements

The authors thank the National Science Foundation for support under award DUE-2122690. All opinions reflected in this paper are those of the authors and not necessarily those of the National Science Foundation.

References

[1] Call for code. `https://callforcode.org/`. Accessed 2022-05-12.

[2] United nations' sustainable goals. `https://sdgs.un.org/goals`. Accessed 2022-05-12.

[3] Margaret E Beier, Michelle H Kim, Ann Saterbak, Veronica Leautaud, Sandra Bishnoi, and Jaqueline M Gilberto. The effect of authentic project-based learning on attitudes and career aspirations in stem. *Journal of Research in Science Teaching*, 56(1):3–23, 2019.

[4] William S Bolton, Shu Ng, Angela Lam, James Kinch, Victor Parchment, William P Foster, Manuela R Zimmermann, Jye Quan Teh, Abigail Simpson, Karisma Sharma, et al. Virtual hackathon to tackle covid-19 unmet needs. *BMJ Innovations*, 7(2), 2021.

[5] Geneva Gay. *Culturally responsive teaching: Theory, research, and practice*. teachers college press, 2018.

[6] David Gibson, Nathaniel Ostashewski, Kim Flintoff, Sheryl Grant, and Erin Knight. Digital badges in education. *Education and Information Technologies*, 20(2):403–410, 2015.

[7] Cindy E Hmelo-Silver, Ravit Golan Duncan, and Clark A Chinn. Scaffolding and achievement in problem-based and inquiry learning: a response to kirschner, sweller, and. *Educational psychologist*, 42(2):99–107, 2007.

[8] Jari Jussila, Jukka Raitanen, Anu Helena Suominen, and Anne-Mari Järvenpää. Virtual hackathons—a novel approach for university-industry collaboration. In *The International Research & Innovation Forum*, pages 247–257. Springer, 2020.

[9] Nicolai Krüger and Frank Teuteberg. Are hackathons the new e-learning environments? on the outcome of the# wirvsvirus hackathon from a learning outcome perspective. In *2020 Sixth International Conference on e-Learning (econf)*, pages 225–228. IEEE, 2020.

[10] Lia Buarque de Macedo Guimarães, Rosimeire Sedrez Bitencourt, Camilla Buttura Chrusciak, Marcia Gemari Derenevich, Cristoffer Rodrigues Poncini, Maria Lucia Miyake Okumura, and Osiris Canciglieri Junior. Sustainability hackathon: Integrating academia and companies for finding solutions for socio-environmental problems. In *Integrating Social Responsibility and Sustainable Development*, pages 591–607. Springer, 2021.

[11] Julie P Martin, Nathan Hyungsok Choe, Jared Halter, Margaret Foster, Jeffrey Froyd, Maura Borrego, and Erica R Winterer. Interventions supporting baccalaureate achievement of latinx stem students matriculating at 2-year institutions: A systematic review. *Journal of Research in Science Teaching*, 56(4):440–464, 2019.

[12] Lukas McIntosh and Caroline D Hardin. Do hackathon projects change the world? an empirical analysis of github repositories. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, pages 879–885, 2021.

[13] Amynah S Mevawala, Faith A Strunk, Roya Haghiri-Vijeh, Inge B Corless, Padmavathy Ramaswamy, Kendra J Kamp, Sheryl Scott, and Sarah Gray. Scientific global nursing hackathon experience. *Nurse Educator*, 46(6):E154–E157, 2021.

[14] Kristie J Newton, Jacqueline Leonard, Alan Buss, Christopher G Wright, and Joy Barnes-Johnson. Informal stem: Learning with robotics and game design in an urban context. *Journal of Research on Technology in Education*, 52(2):129–147, 2020.

[15] Jillian Powers and Mohammad Azhar. Preparing teachers to engage students in computational thinking through an introductory robot design activity. *Journal of Computers in Mathematics and Science Teaching*, 39(1):49–70, 2020.

[16] Chris Rennick, Gregory Litster, Carol CW Hulls, and Ada Hurst. Curricular hackathons for engineering design learning: The case of engineering design days. 2022.

[17] Brandon Reynante. Learning to design for social justice in community-engaged engineering. *Journal of Engineering Education*, 111(2):338–356, 2022.

[18] Caio Steglich, Sabrina Marczak, Luiz Guerra, Cássio Trindade, Alessandra Dutra, and Ana Bacelo. An online educational hackathon to foster professional skills and intense collaboration on software engineering students. In *Brazilian Symposium on Software Engineering*, pages 388–397, 2021.

[19] Katherine N Vela, Rachelle M Pedersen, and Macie N Baucum. Improving perceptions of stem careers through informal learning environments. *Journal of Research in Innovative Teaching & Learning*, 13(1):103–113, 2020.

[20] Shan Wang, William Yeoh, Jie Ren, and Alvin Lee. Learnings and implications of virtual hackathon. *Journal of Computer Information Systems*, 62(3):547–559, 2022.

[21] Jeannette M Wing. Computational thinking. *Communications of the ACM*, 49(3):33–35, 2006.

[22] Jeannette M Wing. Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1881):3717–3725, 2008.

[23] Kevin Kam Fung Yuen and Amy Ooi Mei Wong. Designing an effective hackathon via university-industry collaboration for data science education. In *2021 IEEE International Conference on Engineering, Technology & Education (TALE)*, pages 1–6. IEEE, 2021.

Trustworthiness of Artificial Intelligence: Teaching Factors that Influence AI-supported Decision Making*

Sambit Bhattacharya, Bogdan Czejdo
Department of Mathematics and Computer Science
Fayetteville State University
Fayetteville, NC 28301
{sbhattac, bczejdo}@uncfsu.edu

Abstract

Reports of progress in research into Artificial Intelligence (AI) and its applications are accumulating very rapidly. Specifically, Machine Learning (ML) applications based on large data sets have moved to the forefront of innovations in the field. New ML models have led to the adoption of AI in different disciplines. The development of the most recent large language models has created so much interest that it might mean a revolution in using of AI. Some make a stronger claim that it is a turning point in human civilization's history, and we have started the AI age after replacing the obsolete in many aspects of Information Age applications. One of the immediate challenges is how to use AI and ML responsibly with proper protection for humans and human society. In this paper, we report on our efforts in introducing trustworthiness of ML in college curricula and what factors influence AI-supported decision making. The main goal is to allow students to gain an understanding not only of concepts but also of the limitations of AI. This will help in their participation in our society of the AI Age. The process of AI democratization needs to be established to control the growth of ML use and understand the human dangers of various types of data-driven modeling approaches in AI.

1 Introduction

For the last few years, we observed enormous and very successful research in Artificial Intelligence (AI), specifically Machine Learning (ML), using large data sets. The created ML models lead to the adoption of AI in different disciplines. The US Government has also prioritized AI [2] and is launching efforts that will impact society and people [9]. One of the important efforts is to identify situations where the actions of AI in the real world were of major concern [7]. Ethical problems involving data and privacy with unintended memorization of sensitive data in a neural network [8], bias in financial services [17], and the arrest of an innocent man caused by facial recognition software [6] strongly suggest that current frameworks are inadequate, perhaps very wrong, and can sometimes cause more harm than good [20].

For several years researchers have predicted that at some moment "Deep learning is going to be able to do everything" [14]. In 2022 this prediction gained more credibility with the development of large language models (LLMs) better aligned with human intent the most well-known of which is InstructGPT [22]. ChatGPT which interacts with humans in a conversational way was released later by OpenAI as a sibling model to InstructGPT and this breakthrough created so much interest that it may mean we are now entering the AI Age after the Information Age. ChatGPT allows the generation variety of stories, essays, evaluations of these essays and much more [19]. Practically any activity based on written human communication can be now affected. With fundamental enhancements in ML functionality, there will be a dramatic increase in many related applications [1].

Following ChatGPT's release the use of older methods like filtering and moral/ethics-based model training have been found inadequate with the newest developments. Experimenters have shown how easy is to circumvent GPT3.5 learned "moral compass" [21] while attempts are made to continuously build applications that may help humans in meaningful and profound ways [11]. In addition, image-generating AI has significantly improved [12] and the new models are allowing users to specify instructions written in natural language to produce very realistic pictures reflecting the described situation. As an example, you can upload some real pictures of anybody and change them with instructions written in plain language, to produce believable portraits in any way you want including those which can be used in a malicious way.

Our efforts described in this paper are to allow students to get an understanding not only of concepts but also of the limitations of AI. This will help in their participation in the society of the AI age. The process of AI democratization needs to be established to control the growth of ML use and understand the human dangers of various types of data-driven modeling approaches in AI. Our efforts currently target three groups of students: computer science, forensics, and criminal justice. In the future, we plan to extend the range of participants, including college and university faculty members from different disciplines at 2-and-4 year institutions.

Our project foundations are consistent with the studies that show that student motivation and abilities in computational thinking can be positively impacted by including applications that are relevant to society. This project adds to the growing body of knowledge on how to motivate the study of AI, how to prepare students for the modern-day workplace where both AI skills and domain knowledge are important, and how to train informed and socially responsible creators and users of AI.

2 New AI Era

Recent progress in Artificial Intelligence (AI), the digital transformation of work and human communication has created a wide range of possible applications of AI. Humanity's capacity to generate, store and process data is rapidly increasing, and Big Data is creating new opportunities to find answers to questions and make predictions. Almost every area of human knowledge from the sciences and engineering, humanities and the arts, can be potentially enhanced by AI, and this progress especially with LLMs has raised philosophical questions which are also very practical [25]. While it will take more time and effort to realize its full potential these areas, the risks of adopting AI is an urgent and important problem because it can potentially cause harm when it is prematurely deployed before being tested for trustworthiness [23]. Unlike traditional algorithms and software processes, AI research creates predictive models that are commonly described as "black boxes" since their inner workings are not always open to scrutiny [13]. AI models are trained by using data-driven, bottom-up processes to detect patterns in data instead of being programmed with decision-making logic in a top-down fashion, and this is what makes AI models opaque. It is extremely important that these models are trustworthy since they impact lives, and it is now apparent that increasing the trustworthiness of models is a multi-disciplinary effort that must include domain knowledge from areas that have little in common with AI and Computer Science [5, 24]. The main theme of this paper is that it is imperative that we train the future generation of workers in AI and along with that, the knowledge of its limitations in order to create a more fair and just society.

3 Trustworthiness of AI

Recent progress in Artificial Intelligence (AI), the digital transformation of work and human communication has created a wide range of new applications of AI. Various aspects of trustworthiness of AI are described in the literature [16]. Based on the literature we synthesized the definition of trustworthiness as a four-dimensional concept with the dimensions of safety, fairness, interpretability, and alignment with humanity goals. These dimensions can be described in a general diagram in Fig. 1 showing their relations with social changes caused by Computer Systems. The appearance of Computer Systems has influenced and continue to influence social changes. There are and will be some negative consequences but for the Information Era they were relatively easily identified and alleviated. The more difficult problem is to control negative aspects of AI applications based on ML. As shown in Fig. 1 we can analyze social changes caused by AI using three angles. The first angle is to analyze AI assistance in any aspect of human vs human interactions, the second angle underlines AI machines vs. human situations, and the third angle is to predict the future environment with prevailing AI presence.

In Fig. 1 indicates that for the "human vs. human" angle of analysis all four dimensions can play important role. For the environment including AI machines the main issue is safety i.e., whether AI based machines would not cause any harm for humans. For the "future" environment stressing the need for future consideration is critical. AI changes are so quick that neglecting the analysis of predicted applications may cause the irreparable damage for the human civilization.

4 Curricular Modules for Infusion of Trustworthiness of AI

Trustworthiness of AI are addressed with respect to the specific AI System used for the specific task. In the first curricular module we start with typical analysis of a particular AI system i.e., dataset creation, model training, and model performance evaluation metrics as indicated in the column 1 of the Trustworthiness Analysis Table in Fig.2. Next, we need to analyze the specific task for which the AI system is applied including constraints and requirements as shown in the column 2 of the Trustworthiness Table. In order to determine safety aspect of the trustworthiness in using the AI System for the application task, we need to list the safety expectations as described by the AI users. Safety of AI system is discussed in the literature [4] very extensively. Various levels of safety are proposed including unsafe ("harming humans"), conditionally safe, and generally safe. The more specific estimation of safety can be based on some safety margins e.g., for autonomous driving. These expectations can be expressed verbally, but we emphasize the importance of specifying numeric margins, if possible, describing the acceptable metrics of ML performance as indicated in the column 3.1 of the Trustworthiness Analysis Table.

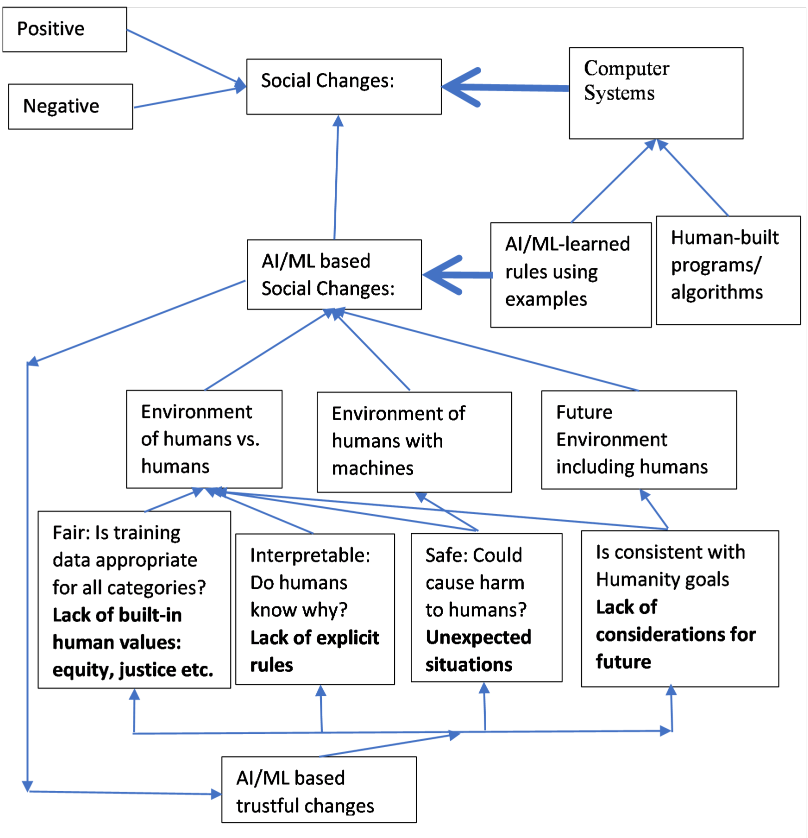


Figure 1: Four Dimensions of Trustworthiness

1. AI System	**2. Application Area Task**	**3. Trustfulness for AI System with respect to the Application Task**			
		3.1 Safety	**3.2 Fairness**	**3.3 Interpretability**	**3.4 Alignment**
Model and Performance	**Description, Constraints, Requirements**	**Safety Margins**	**Fairness Margins**	**Interpretability**	**Alignment with humanity goals and values**
Recognize a human face from a set of faces; Neural Networks; Performance Metrics Accuracy 99.9%	Compare crime scene pictures (from video) with the set of pictures, identify crime suspect and make an arrest	If safety margins are [99.5%,100%] the application is safe	If fairness margins for accuracy are [-1%,+1%] as Fig.1b. shows fairness is not satisfied	Neural Network as a ML Model – no interpretability. LIME approach can provide partial interpretation.	Conditionally OK is fairness is satisfied but cannot be extended to some other tasks

Figure 2: Trustworthiness Analysis Table

We call AI system fair with respect to the related application task if it performs well and fairly for all subcategories of data. Typically, the fairness requirements are related to various subcategories of data selected on the base of demographic factors (e.g., age, sex, ethnicity), but generally the users of AI system need to determine also any other sensitive factors. These expectations can be again expressed verbally but we encourage specification of a numeric margins describing the maximum allowed difference in model performance metrics for the various subcategories of data. The Fairness Tree can be used to describe whether safety margins are satisfied for each sensitive categorization.

Let us review these two dimensions of trustworthiness using some synthetic data for the example of AI system used for the face recognition [3]. The summaries in last row of the Trustworthiness Analysis Table in Fig.2 indicate that the safety margins of accuracy metrics [99%,100%] are satisfied. The AI system is, however, not fair because it does not treat different categories of race; one category "Black" has accuracy almost 10% lower. By teaching our students to employ our analysis we can help them be aware of such shortcomings and avoid the serious and unexpected problems in their professional life.

With the diversity of machine learning models and their applications there is a need to support trustworthiness of AI system users by better interpretation and visualization of the results. Historically, the ML models based on Decision Trees, especially with some simplified versions, could help understand the automatic decision process. Recently, with overwhelming majority of ML models based on neural networks a new approach to interpretation and visualization of machine learning process is necessary. Therefore, we include the training in techniques such as Local Interpretable Model-Agnostic Explanations (LIME)

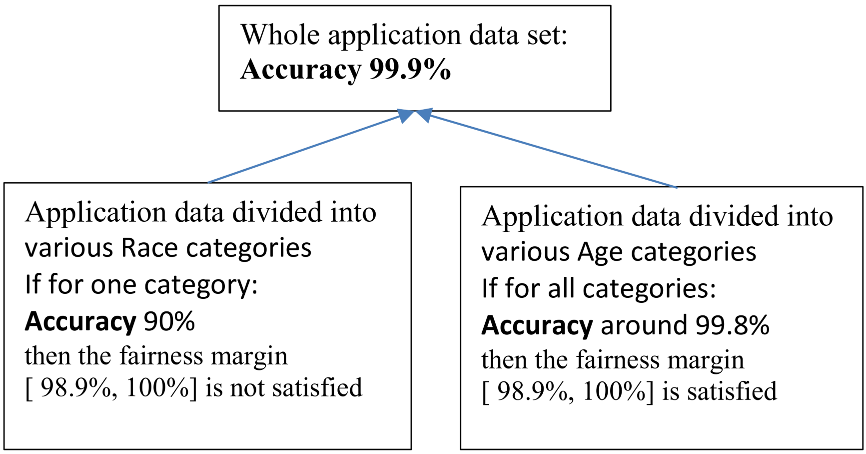


Figure 3: Fairness Tree

[10] that is based on perturbing data and finding its implications as shown in column 3.3 in Trustworthiness Analysis Table. The main advantage of this approach is to increase trust in the AI solutions by employing human in the loop to provide the feedback to the whole process of the AI guided task. In our example of face recognition some advanced techniques are necessary for perturbing the pixel data but in other cases with numeric data the LIME experiments are straightforward.

The fourth dimension called alignment reflects whether the specific AI application leads into a direction that is aligned with humanity goals and its values [16]. Our educational activities trigger multi-disciplinary discussions about the current state and the future of humanity. They include futuristic applications of artificial general intelligence (AGI). For our example, described in column 3.4 of the Trustworthiness Analysis Table our students get heavily involved in the discussion where the use of AI to recognize faces can be acceptable. That can also revive the discussion about the relationship between safety and fairness margins and their possible values in the context of humanity goals and values.

The described project creates educational modules that infuse the knowledge and skills of determining trustworthiness of AI in courses that currently lack this content. It builds upon a previous effort of organizing a Workshop for AI concepts [our Workshop paper]. The teaching method of Process Oriented Guided Inquiry Learning (POGIL) is adopted to create carefully designed sequences of inquiry- based tasks that are being completed by students in teams [26, 15, 18]. Most of these tasks involve hands-on, experimental work with AI models that execute within a sandbox software environment. Through immersion in this process students learn how AI models work and how to assess trust, which is essential when taking actions based on the predictions of models, or when deploying a new model. This guided process also provides insights into AI models, which students can use to transform an untrustworthy model or prediction into a trustworthy one. The project creates educational materials that address the social relevance of AI which is a topic that is not emphasized in current computing curricula.

5 Summary

The described project is based on AI research and educational research that suggest computing and non-computing curricula can benefit from customized modules that expose students to the knowledge of AI and the trustworthiness of AI models. The trustworthiness and social relevance of AI is a profoundly multi-disciplinary effort. This project supports the adoption of AI in non-computing disciplines and strengthens computing curricula with content that has traditionally not received adequate coverage. The project educates undergraduate students from computing and non-computing majors about how AI advances the goals of society and at the same time raises very important concerns related to trustworthiness. Students impacted by this work gain deeper knowledge of multi-disciplinary efforts to improve computing as an important tool for society. AI technologies have experienced robust job growth recently and this trend is expected to continue. The project develops capacity at a primarily undergraduate and historically black university, and engage students, with focus on students from underrepresented communities, to work at the convergence of social issues and computing. Non-computing graduates seeking employment also increases their competitiveness in the job market by understanding the potential of AI and its limitations, and they will be equipped with tools and the understanding of how to use AI responsibly within their own disciplines.

References

[1] ChatGPT: Optimizing language models for dialogue (OpenAI). https://openai.com/blog/chatgpt/. Accessed 2022-12-10.

[2] Executive order on AI (US Government). https://www.whitehouse.gov/ai/. Accessed 2020-11-12.

[3] B. Allyn. 'The computer got it wrong': How facial recognition led to false arrest of black man. https://www.npr.org/2020/06/24/882683463/the-computer-

`got-it-wrong-how-facial-recognition-led-to-a-false-arrest-in-michig`. Accessed 2022-12-10.

[4] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*, 2016.

[5] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information fusion*, 58:82–115, 2020.

[6] TJ Benedict. The computer got it wrong: Facial recognition technology and establishing probable cause to arrest. *Wash. & Lee L. Rev.*, 79:849, 2022.

[7] Richard Berk. *Criminal justice forecasts of risk: A machine learning approach.* Springer Science & Business Media, 2012.

[8] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *USENIX Security Symposium*, volume 267, 2019.

[9] Melissa M Chen, Lauren Parks Golding, and Gregory N Nicola. Who will pay for AI? *Radiology: Artificial Intelligence*, 3(3):e210030, 2021.

[10] Bogdan Czejdo, Sambit Bhattacharya, and Catherine Spooner. Improvement of protein model scoring using grouping and interpreter for machine learning. In *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 0349–0353. IEEE, 2019.

[11] Cami Czejdo and Sambit Bhattacharya. Towards language models for AI mental health assistant design. In *2021 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 1217–1222. IEEE, 2021.

[12] Mohamed Elasri, Omar Elharrouss, Somaya Al-Maadeed, and Hamid Tairi. Image generation: A review. *Neural Processing Letters*, 54(5):4609–4646, 2022.

[13] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5):1–42, 2018.

[14] Karen Hao. AI pioneer "geoff hinton":“deep learning is going to be able to do everything”, 2020.

[15] Helen H Hu, Clifton Kussmaul, Brian Knaeble, Chris Mayfield, and Aman Yadav. Results from a survey of faculty adoption of process oriented guided inquiry learning (POGIL) in computer science. In *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education*, pages 186–191, 2016.

[16] Sonali Jain, Manan Luthra, Shagun Sharma, and Mehtab Fatima. Trustworthiness of artificial intelligence. In *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, pages 907–912. IEEE, 2020.

[17] Aaron Klein. Reducing bias in AI-based financial services. brookings report, 2020.

[18] Clifton Kussmaul, Helen H Hu, and Matthew Lang. Guiding students to discover CS concepts and develop process skills using POGIL. In *Proceedings of the 45th ACM technical symposium on Computer science education*, pages 745–745, 2014.

[19] Stephen Marche. The college essay is dead. `https://www.theatlantic.com/technology/archive/2022/12/chatgpt-ai-writing-college-student-essays/672371/`. Accessed 2022-12-10.

[20] Brent Mittelstadt. Principles alone cannot guarantee ethical AI. *Nature machine intelligence*, 1(11):501–507, 2019.

[21] Zvi Mowshowitz. Jailbreaking ChatGPT on release day. `https://thezvi.substack.com/p/jailbreaking-the-chatgpt-on-release`. Accessed 2022-12-10.

[22] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*, 2022.

[23] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should I trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[24] Ribana Roscher, Bastian Bohn, Marco F Duarte, and Jochen Garcke. Explainable machine learning for scientific insights and discoveries. *IEEE Access*, 8:42200–42216, 2020.

[25] Murray Shanahan. Talking about large language models. *arXiv preprint arXiv:2212.03551*, 2022.

[26] Paulette Vincent-Ruz, Tara Meyer, Sean G Roe, and Christian D Schunn. Short-term and long-term effects of POGIL in a large-enrollment general chemistry course. *Journal of Chemical Education*, 97(5):1228–1238, 2020.

Learning Lists and Dictionaries by Building Web Dashboards with Live Data*

Nifty Assignment

Eric D. Manley
Department of Mathematics and Computer Science
Drake University
Des Moines, IA 50311
eric.manley@drake.edu

This Nifty Assignment tasks students with creating an interactive, web-based data visualization dashboard connected to a live data source. The assignment is targeted for late-CS1/early-CS2 after students have experience with loops and know how to access individual items in lists and dictionaries. Students are walked through the use of Python libraries for making HTTP data request, creating visualizations, and integrating them into a web framework with user interface components. The challenge of the assignment comes in working with the received data, containing nested lists and dictionaries. Students learn to explore the data, selecting needed fields as well as iterating through and filtering. Overall, there are several benefits of utilizing these tools:

- students get experience working with real-world, non-trivial examples of nested lists and dictionaries
- the application students develop is visually appealing, fun to use, and can be deployed and shared
- students get a taste of how to build professional-quality web applications
- the data visualization aspect provides early exposure to data science work for students who might be interested in that area
- the assignment serves as the springboard for many creative projects through substitution of different data sources and visualization types

An example of an end-result of the assignment be seen in Figure 1. Assignment materials can be found at

Figure 1: an example of dashboard resulting from the assignment

`http://analytics.drake.edu/~manley/nifty/NiftyCOVIDDashboard.html`

The assignment is presented in three parts, covering the three components of the application, and instructors could choose to use one, two, or all three parts.

Part 1: Web APIs: Part 1 introduces the Python `requests` library for making HTTP requests to Web APIs. We utilize a free COVID-19 API [3], which is easy-to-use and does not require any authentication. There are a number of different endpoints that developers can use, and the JSON data is returned as a Python object containing nested lists and dictionaries.

Part 2: Plotly Visualizations: Plotly [2] is an open-source Python library for creating interactive data visualizations. It includes all major types of charts (line, bar, scatter, etc.) with many features like mouse hover effects. Many kinds of charts recognize data in the list-of-dictionary-records format (making it perfect for the data from Part 1) and can be generated with one line of code. To maximize the usefulness of these visualizations, though, students will first get to do some processing of their data.

Part 3: Dash Application: Dash [1] is an open-source framework enabling rapid development of web applications. Dash applications sit on top of a Flask server and generates React front ends. Developers need not know any HTML, JavaScript, Flask, or React. Getting started is just as easy as other Python GUI frameworks like `tkinter`. Dash was developed by the Plotly company and by design works seamlessly with Plotly visualizations, so students can easily build a web UI around their work from Part 2. The assignment materials include an optional lab that can get students up-to-speed with Dash before using it in the assignment.

References

[1] Dash Python user guide. `https://dash.plotly.com/`. Accessed 2022-11-17.

[2] Plotly open source graphing library for Python. `https://plotly.com/python/`. Accessed 2022-11-17.

[3] Kyle Redelinghuys. COVID 19 API. `https://covid19api.com/`. Accessed 2022-11-17.

Online Networking and Security Project*

Nifty Assignment

Wen-Jung Hsin
Computer Science and Information Systems
Park University
Parkville, MO 64152

In learning computer networking and security concepts, it is best accompanied by hands-on projects using physical devices. Typically, a physical networking and security laboratory can only be used for face-to-face in-person learning. For distant students, physically operating the actual devices is not possible. At our university, our distant students perform experiments using an online hands-on networking and security laboratory with physical devices, called Netlab+[1]. In this proposal, we present a hands-on project for the distant students to perform in the Netlab+ environment.

Physical Networking and Security Project

Objective:

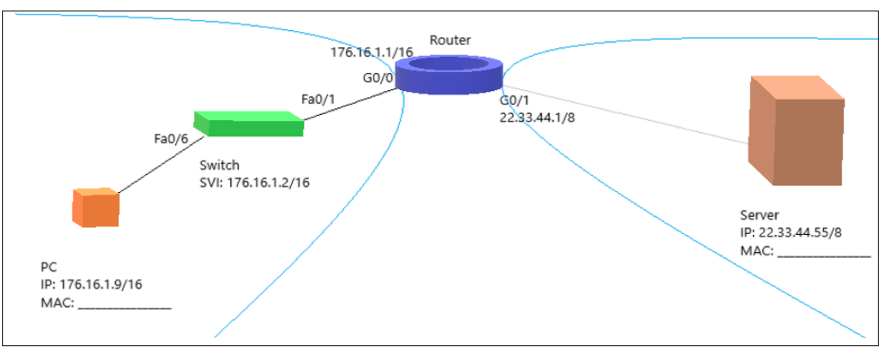


Figure 1: The Network

Establish and secure network connectivity using Wireshark, Switch Port-based Security, and Router Packet Filtering via online Netlab+ [1] laboratory.

Procedure:

Initial:

Given Figure 1, 1 router, 1 switch, 1 server, and 1 PC have already been cabled accordingly.

Step 1: Establish the connectivity

Configure the required IP addresses for all devices accordingly.

Issue ***ipconfig /all*** on both PC and Server. Record the MAC addresses of PC and Server in Figure 1.

Issue ***pings*** between PC and Server. Make sure that all pings are successful. Screen capture the ping results.

Step 2: Use Wireshark to examine ICMP packet

Start Wireshark capture on PC.

Issue a ping from PC to Server.

Locate an ICMP packet sent from PC to Server in Wireshark. Does the IP and MAC addresses of the ICMP packet in Wireshark match the recorded addresses in Figure 1? Yes or no. Explain why or why not. Screen capture and highlight the addresses shown in the Wireshark screenshot.

Step 3: Examine the Mac Address Table in Switch

In Switch, issue ***clear mac address-table dynamic***, then ***show mac address-table***.

Issue ***ping*** from PC to Server.

In Switch, issue ***show mac address-table***. Do the addresses shown in the mac address table match the ones in Figure 1? Yes or No. Screen capture the mac address table.

Step 4: Secure Switch Port

Configure switch port Fa0/6 to allow only one specific MAC address of 1111.1111.1111. This particular address is different from PC's MAC address.

Issue ***ping*** from PC to Server. Is the ping successful? Yes or No. Explain why or why not.

What to Submit:

Submit your answers for all questions and screenshots from all steps.

References

[1] Netlab+. (Network Development Group, Inc.). `https://www.netdevgroup.com/products/`. Accessed 2023-01-27.

Mathematics Appreciation: Golden Ratio*

Nifty Assignment

Wen-Jung Hsin
Computer Science and Information Systems
Park University
Parkville, MO 64152

Since its discovery by mathematician Euclid 300 B.C. [2], Golden Ratio has continued to show its prominent value in mathematics, science, arts, nature, and more. To cultivate and elicit appreciation from the students, the proposed assignment encourages the students to be creative in discovering the things around them in their daily lives that exhibit the golden ratio.

First, what is the golden ratio? Given two numbers x and y, where y is greater than x, the ratio $\frac{y}{x}$ is the golden ratio when $\frac{y}{x} = \frac{x+y}{y}$. Golden ratio, denoted by symbol ϕ, has an irrational value of 1.618 approximately. In Discrete Mathematics, the teaching in recursion naturally leads to a prominent example, Fibonacci Sequence, which is 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ... where each number is the sum of the previous two numbers. In Fibonacci sequence, when a number is divided by its previous number in sequence, the corresponding ratio sequence is 1, 2, 1.5, 1.666, 1.600 1.625, 1.615, 1.619, 1.617, 1.618, 1.618, ... which approaches ever closer to ϕ, the Golden Ratio. Typically, in Discrete Mathematics course, we assign students to perform various logic thinking and computation. The following is a light, refreshing assignment, giving students a breather from heavy computing.

1 Assignment: Golden Ratio Around Me

1.1 Objective

Discover various objects or phenomena in your daily life that exhibit the golden ratio.

Grading Rubric – Golden Ratio Project

Criteria	Rating				Pts
Application	3 pts Full Mark is original, interesting, creative, and inspiring	2 pts Almost There Mark Lacks one of the following attributes: original, interesting, creative, inspiring	1 pts Partial Mark Lacks two of the following attributes: original, interesting, creative, inspiring	0 pts No Mark Significantly Lacks the following attributes: original, interesting, creative, and inspiring	3 pts
Reflection Writing	3 pts Full Mark is original, interesting, creative, and inspiring	2 pts Almost There Mark Lacks one of the following attributes: original, interesting, creative, inspiring	1 pts Partial Mark Lacks two of the following attributes: original, interesting, creative, inspiring	0 pts No Mark Significantly Lacks the following attributes: original, interesting, creative, and inspiring	3 pts
				Total	6 pts

Figure 1: Rubric

1.2 Steps

1. Watch a YouTube video entitled "Golden Ratio = Mind Blown!" [1] in which the speaker enthusiastically shows the objects and phenomena exhibiting the golden ratio.

2. Find two or more things in your daily life that exhibits the golden ratio.

3. Take a picture of each item you find.

4. Edit each picture to show the golden ratio in the picture.

5. Reflection: Write a minimum of 300-word short essay reflecting on what you learn from the golden ratio, what the golden ratio entails, or what your impression is about the golden ratio. For example, after learning about the golden ratio, do you have more appreciation of mathematics than before?

1.3 Grading

The grading is based on whether the work is original, interesting, creative, and inspiring. See Figure 1 for the grading rubric.

References

[1] Golden ratio = mind blown. https://www.youtube.com/watch?v=c8ccsE_IumM. Accessed 2023-01-27.

[2] An introduction to the golden ratio. https://www.adobe.com/creativecloud/design/discover/golden-ratio.html. Accessed 2023-01-27.

Dash: An Easy-to-Use Framework for Building Web Applications and Dashboards*

Conference Workshop

Eric D. Manley
Department of Mathematics and Computer Science
Drake University
Des Moines, IA 50311
eric.manley@drake.edu

This workshop will introduce Dash, an open-source framework that enables rapid development of Python web applications [1]. Dash has recently been gaining popularity as a tool for building professional-quality data visualization dashboards; however, because it includes a full range of user interface components (buttons, dropdown menus, sliders, canvases, etc.) it can also be used to build almost any kind of web-based application. Dash applications generate HTML and React.js, though because of the low-code design approach, no knowledge of these is necessary; and very little code is needed to get up and running. Furthermore, callback functions are designated using intuitive function decorators which make it easy to see exactly what should trigger a callback to run and what should be changed as a result. All of this makes it an accessible option for CS1/2 students being introduced to graphical user interfaces. Furthermore, Dash integrates seamlessly with the Plotly data visualization library [4], and so it is an excellent opportunity to introduce data science use cases in early programming courses. There are also many options for students to deploy and share their Dash applications, including some free and easy to use services like Heroku [3] and Python Anywhere [5], because it sits on top of a Flask web server [2], which is a widely supported Python web framework.

As part of the workshop, we will cover

- building applications with basic UI components

- writing callback functions, including those with multiple triggers and outputs as well as chained callbacks
- integrating Plotly graphs and maps into Dash applications
- a discussion of considerations necessary to deploy applications publicly on the web

References

[1] Dash Python user guide. `https://dash.plotly.com/`. Accessed 2022-11-17.

[2] Flask: web development one drop at a time. `https://flask.palletsprojects.com`. Accessed 2022-11-21.

[3] Heroku cloud application platform. `https://www.heroku.com/`. Accessed 2022-11-21.

[4] Plotly open source graphing library for Python. `https://plotly.com/python/`. Accessed 2022-11-17.

[5] Python anywhere. `https://www.pythonanywhere.com`. Accessed 2022-11-21.