The Journal of Computing Sciences in Colleges

Papers of the 32nd Annual CCSC Mid West Conference

September 19th-20th, 2025
Wabash College
Crawfordsville, IN

Abbas Attarwala, Editor
California State University, Chico

William Turner, Regional Editor
Wabash College, Crawfordsville, IN

Volume 41, Number 4

September 2025

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

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.

Gold Level Partner

Rephactor
ACM2Y
Blossoms

Welcome to the 2025 CCSC Mid West Conference

Welcome to the 32nd Annual Midwest Conference at the Wabash College in Crawfordsville, IN. This year's conference brings educators, researchers, students, industry professionals, and industry partners from across the region to explore the latest advancements and share insights in computer science. Sessions include a pre-conference workshop, keynote and dinner speakers, refereed papers, panels, tutorials, nifty assignments, works in progress, vendor talks, student showcase, and a student hackathon programming contest.

We accepted 12 of 18 excellent paper submissions, a 66% acceptance rate. These selected papers represent cutting-edge research and innovative practices across various topics. We look forward to the pre-conference workshop by Cathy Bareiss, covering program and course assessment.

We are excited to feature Clark Cully as our keynote speaker, who will discuss Computer Scientists as Innovators. We are also honored to have Jim Huggins as our banquet speaker, sharing his top stolen teaching tips.

We extend our heartfelt thanks to everyone who made this conference possible: the conference committee, paper reviewers, speakers, presenters, and especially our site chairs, Colin McKinney and William Turner, and Wabash College for their exceptional support in hosting this year's event. We also thank our National Partners ACM2Y, Rephactor, and Blossoms for their continued support of our activities and UPE for student prizes.

We have a full schedule of events and encourage you to make the most of the many opportunities for learning, collaboration, and professional growth. Your participation is critical to the success of this conference, and we hope you find the sessions engaging and inspiring.

Thank you for joining us, and we look forward to a productive and enjoyable conference!

Paul Talaga
University of Indianapolis
2025 Midwest Conference Chair

2025 Steering Committee - CCSC Midwestern Region

Deborah Hwang, Registrar (2025) .... University of Evansville, Evansville, IN
William Turner, Editor (2027) .... Wabash College, Crawfordsville, IN
Michael P. Rogers, Webmaster (2025) ..... University of Wisconsin Oshkosh, Oshkosh, WI
Dominic Wilson, Treasurer (2026) .... University of Findlay, Findlay, OH
Karl Schmitt, At-Large (2026) ... Trinity Christian College, Palos Heights, IL
Jeff Lehman, At-Large (2027) .... Huntington University, Huntington, IN
David Largent, Regional Representative (2026) Ball State University, Muncie, IN
Paul Talaga, Conference Chair (2025) .... University of Indianapolis, Indianapolis, IN
Jeff Lehman, Past Conference Chair . Huntington University, Huntington, IN

2025 CCSC Midwestern Conference Committee

Paul Talaga, Conference Chair (2025) .... University of Indianapolis, Indianapolis, IN
Zachary Kurmas, Vice Chair ... Grand Valley State University, Allendale, MI
Colin McKinney, Site Chair .... Wabash College, Crawfordsville, IN
William Turner, Assistant Site Chair .... Wabash College, Crawfordsville, IN
Imad Al Saeed, Papers .... Saint Xavier University, Orland Park, IL
Ahmed Elmagrous, Nifty Tools & Assignments .... University of Wisconsin-Stout, Menomonie, WI
Fola Olagbemi, Assistant Nifty Tools & Assignments . Hope College, Holland, MI
Ahmed Elmagrous, WIP ..... University of Wisconsin-Stout, Menomonie, WI
Sugandha Malviya, Panels, Tutorials, Workshops ...... Ball State University, Muncie, IN
Cathy Bareiss, Assistant Panels, Tutorials, Workshops .... Bethel University, Mishawaka, IN
William Turner, Authors (2027) .... Wabash College, Crawfordsville, IN
Jeff Lehman, Past Conference Chair . Huntington University, Huntington, IN
Paul Talaga, Programming Contest Chair .... University of Indianapolis, Indianapolis, IN
Md Haque, Programming Contest (Assistant) .... University of Indianapolis, Indianapolis, IN
Nathan Sommer, Programming Contest (Assistant) . Saint Xavier University, Orland Park, IL
David Largent, Publicity .... Ball State University, Muncie, IN
Deborah Hwang, Registrar (2025) .... University of Evansville, Evansville, IN

Stefan Brandle, Speakers Chair .... Taylor University, Upland, IN
Andy Harris, Student Showcase Chair ..... Ball State University, Muncie, IN
Dominic Wilson, Treasurer (2026) .... University of Findlay, Findlay, OH
Kris Roberts, Two-year College Liaison .. Ivy Tech Community College, Fort Wayne, IN
Takako Soma, Vendors .... Illinois College, Jacksonville, IL
Cathy Bareiss, Virtual Host .... Bethel University, Mishawaka, IN
Michael P. Rogers, Webmaster (2025) ..... University of Wisconsin Oshkosh, Oshkosh, WI
William Turner, Assistant Webmaster ... Wabash College, Crawfordsville, IN

Good Artists Copy, Great Artists Steal: My Top Stolen Teaching Tips*

Evening Address

Jim Huggins, Kettering University, Flint, MI

An old aphorism asserts: "we see further than others because we stand on the shoulders of giants". As I reflect upon my teaching career, I realize that much of my best pedagogy (and perhaps my best research!) comes not from original thoughts, but ideas I've adopted from mentors and colleagues over the years. In this evening address, I'll share some of my best "stolen" teaching tips, in the hopes that others can benefit from them just as I did.

Dr. James K. (Jim) Huggins is an associate professor of computer science at Kettering University in Flint, Michigan. Jim received his Ph.D. in computer science from the University of Michigan in 1995, where he also earned his B.S. and M.S. as a first-generation college student. His thesis focused on the application of formal methods to programming languages and systems. Jim joined the faculty of Kettering University (then GMI Engineering & Management Institute) in 1997, where he has served on the faculty for nearly thirty years. Jim's current academic interests including computer science education, computing history, and computing ethics. Jim has been part of the leadership team of the AP Computer Science A Exam Reading for the last fifteen years. Jim received Kettering University's Outstanding Teaching Award in 2002 and 2011.

Computer Scientists as Innovators: A Call for Transformational Leadership*

Keynote

Clark Cully
Technology Division Chair
Indiana Wesleyan University

Given the rapid pace of innovation, many universities and organizations are struggling to embrace AI and other digital tools to re-envision their work. Drawing on insights from leading transformation in the military, government, and academic, Dr. Cully will explore the vital role of computer scientists in bridging the gap in imagination and creating a technology-enabled culture.

Dr. Clark Cully is the former CDO and Chief AI Strategist for the Department of Defense. During his 14 years in the Pentagon, he coauthored military strategies for cyber warfare, data management, nuclear modernization, and digital transformation. He also served as a senior government advisor on topics including deterrence, command & control, emerging technology, and organizational reform.

Dr. Cully is now the Technology Division Chair at Indiana Wesleyan University, overseeing 2500 students studying artificial intelligence, data analytics, information technology, and cybersecurity. Dr. Cully holds a Ph.D. in high-energy physics from the University of Michigan and degrees in mathematics and physics from Calvin University.

Assessment: Beyond the Requirements*

Conference Workshop

Cathy Bareiss
Bethel University
Mishawaka, IN 46545
cbareiss@acm.org

This preconference workshop aims to help those attending to advanced their assessment activities beyond what is required to an activity that provides many benefits with minimum cost. Now that we are all doing assessment, how can we make the most of it without adding a burden to ourselves? This workshop aims to help with that. Let's move assessment from something we **must do** to something we **want to do**.

Rethinking CS Education: How will our students stay competitive in this new AI era? *

Vendor

Emily Wang, Giqi Zhao, Gavin Zhang, and Yi Li
Blossoms
giqi@blossoms.ai

As AI takes over line-by-line coding and even project-level tasks, professors face a pressing question: *What will truly keep our CS students competitive?* This session tackles that challenge through disruption simulations, collaborative activities, and insights from our conversations with over 1,000 CS professors.

We'll open with an interesting disruption exercise that makes the impact of AI on coding skills immediately tangible. From there, participants will engage in hands-on activities and collaborative discussion to uncover which abilities — design, system thinking, critical evaluation, or interdisciplinary application, matter most for CS students today.

We will also provide a brief, accessible introduction to AI interpretability and safety, highlighting why understanding how models behave is now essential in CS education. We'll share real classroom shifts we see already happening: students acting as designers or product managers, critiquing AI-generated solutions, and moving from rote implementation toward tradeoff analysis and system-level thinking. At the conclusion of this session we will introduce **Blossoms** and **Spaceship**, initiatives shaped together with the research community. They reflect our effort to show our vision, our awareness of the challenges, and our persistence in creating something authentic and genuine for CS education.

Introducing State Machines into the Software Engineering Curriculum*

Work in Progress

Ramachandra B. Abhyankar
Indiana State University
RB.Abhyankar@indstate.edu

In the Software Engineering area, Object Orientation(OO), the Unified Modeling Language (UML), the Unified Process (UP) , and Agile Processes, have dominated the curriculum. To a lesser extent, Formal Software Development Approaches like VDM and the B method have also influenced the Software Engineering curriculum. It must be admitted that these methods of software development have not solved the problem they were meant to address: the software crisis. The Software Engineering Education Community needs to take note of another approach to software development, based on "State Machines." The term "state machines" is used here loosely to include related terms such as Finite State Automata, Finite State Machines, Petri-Nets, Extended State Machines and Statecharts. The inspiration for the use of state machines in software development comes from the success state machines have enjoyed in the area of hardware design. It is hoped that that success of state machines can be replicated in software design. Early indications are that that may well be true. Many books and web sites have advocated the use of State Machines in Software Engineering, yet the use of State Machines has not become mainstream in software development. Among the reasons cited are people just do not know about them, and YAGNI (You Ain't Gonna Need It), lack of guidance on implementation techniques in the literature, examples presented in the literature have emphasized hardware examples, not software, etc. I believe another reason is that, currently, the State Machines Approach is not part of the Software Engineering Curriculum. There are many approaches to introducing the State Machines Approach into the Software Engineering Curriculum. These approaches will be discussed.

Computer Science Curriculum Change using Reviews*
Work in Progress

Mohammad Amin, Pradip Peter Dey, Bhaskar Sinha
School of Technology and Engineering
National University
9388 Lightwave Ave., San Diego, CA 92123
{mamin, pdey, bsinha}@nu.edu

Computer Science (CS) curriculum changes in academia may be needed more frequently than we assume. Rapid reviews of any existing curriculum may show opportunities for improvements. Broadly defined criteria for review should be considered in a flexible format without strictly restricting them to narrowly assumed traditional structures. The proposed flexible review framework may bring together interactions among scientific, technological, social, educational, economic, cognitive, and other criteria, so that the changes in curriculum may serve all stakeholders adequately. Many of the criteria mentioned above may imply consideration of curriculum changes in many emerging areas including Quantum Computing (QC). The demand for QC specialists in the industry is rapidly growing, and the progress of QC is faster than it was predicted in the past. Due to this requirement, there is a growing need for academicians to proactively start integrating QC topics into their CS programs to enrich the curricula and make it more relevant to the current market need. This will provide students with the knowledge to address complicated, real-life, and practical problems facing the industry. QC requires a multidisciplinary background to understand the overall concepts, mechanisms, and processing. Universities and colleges can fulfill these requirements by offering courses and training, and certifications in QC at various levels. There are many significant challenges in QC that need to be addressed. These include quantum decoherence and quantum noises, fragile quantum states, error corrections, cryogenic issues, etc. Teaching these QC concepts in academic programs will prepare students for the emerging market demands, encourage interdisciplinary collaborative research, etc., thus leading to augmented problem-solving expertise. Rapid reviews of the current curriculum may suggest the efficacy of QC.

Pilot ABET Accreditation for Associates in IT: Lessons Learned and Guidelines for Prospective Accreditation Applicants

Work-in-progress

Suguna Chundur[1],
[1]Technology and Health Information Systems
University of Cincinnati - Clermont College
Batavia OH 45103
{chundusa,author}@ucmail.uc.edu

Accreditation ensures the level of quality of an educational program. The accreditation provided by the Accreditation Board for Engineering and Technology (ABET) organization "assures confidence that a collegiate program has met standards essential to prepare graduates to enter critical STEM fields" (ABET, 2025). ABET accredits programs at the bachelor's level and above. In the recent past, ABET has started the process of accrediting Associates degree programs in Engineering and Technology. Our associate's in Informa-tion Technology program participated in the Pilot ABET Accreditation cy-cle in 2024-25 and is awaiting the final decision by ABET. This presentation will highlight important aspects of our experience of the pilot accrediting pro-cess. The accreditation experience covers the 18-month process our institution engaged in, from a readiness review to submitting a request for evaluation, followed by completion and submission of a self-study report, preparing for site-visit, culminating with due process response to initial draft report and the final accreditation decision by ABET. Our college is at the final stage, await-ing the accreditation decision by ABET to be made in July. In addition to our experience with the accreditation process, this presentation will also share the important benefits of accreditation such as program recognition by peer institutions and industry, promotion of best practices, self-assessment, quality control and improvement and acceptability of transfer credits.

Anxiety about AI Safety in Learning Environments*
Work in Progress

Pradip Peter Dey, Mohammad Amin, Bhaskar Sinha
School of Technology and Engineering
National University
9388 Lightwave Ave., San Diego, CA 92123
{pdey, mamin, bsinha}@nu.edu

While Artificial Intelligence (AI) might offer valuable capabilities, users and developers may experience anxiety about its safety. In academia, students may have anxiety about AI safety as they study the advantages and disadvantages of emerging AI or Artificial General Intelligence (AGI). The learners may become eager to study AGI issues so that catastrophic harm from AGI can be reduced or avoided. Anxiety may center around questions such as: Is there a way to prevent AGI technology from falling into wrong hands? Will AGI development increase with no upper bound and surpass human intelligence, and dominate over humans? Life begins with playful learning and then continues to deal with increasingly complex issues for meaningful learning with little or no mitigation about anxiety. Teachers may consider team projects for their students that may potentially reduce anxiety. One possible consideration for teachers is to inspire students to work on group projects to design and implement AGI systems that can be used for the benefit of humanity by making sure that AGI's goals are aligned with human values; the final system is intended to become the most powerful among all AGI systems with guaranteed ability to provide protection against harmful agents. Such projects with appropriate support and resources may help reduce anxiety related problems among students. In addition, they may provide strong motivation for students to learn about the range of possible architectural design techniques for the AGI systems with critically thinking about current architectures such as transformers, mixture of experts, and so forth.

Introducing Quantum Concepts, Ethics, and Intergenerational Engagement in Introductory Programming Courses: A Pedagogical Prototype*

Work in Progress

Ladelle S. Augustine Diekhoff
Computer Science Department
Ball State University
ladelle.augustine@bsu.edu

Benjamin Augustine Diekhoff
Computer Science Department
Ball State University
benjamin.diekhoff@bsu.edu

With emerging technologies like quantum computing redefining the digital landscape, there is an urgent need to modernize introductory computer science education. Most introductory courses lack any reference to quantum concepts, making it difficult to prepare students for the future of computing. This proposal presents a forward-thinking curriculum that integrates core programming skills with conceptual introductions to quantum computing framed through real-world challenges such as sustainability, health, and equity. The proceeding highlights include industry predictions; Hazan et al. (2020) cite McKinsey's forecast that the quantum computing market could surpass USD 1 trillion by the mid-2030s, driven by demand in finance, manufacturing, tech, media, and telecommunications. However, students in computer science courses risk being underprepared for roles in these rapidly evolving industries without early exposure. The curriculum accentuates project-based learning, multimedia, and reflective practices to deepen student engagement, civic awareness, and ethical practices. A secondary component of this project is intergenerational outreach, where students work with older adults to understand the benefits of emerging technologies like quantum computing, raising awareness and helping to reduce the digital vulnerability of aging populations. Drawing on Fitzpatrick (2025) work on initiatives that reimagine aging as a positive developmental process central to ethical tech education. The exploration will embark on the inquiry using a mixed-methods approach that combines quantitative metrics (e.g., knowledge gains, engagement levels) with qualitative insights (e.g., student reflections and interviews). Participants will explore practical strategies for embedding emerging technologies and social impact themes into early computer science instruction. The findings will inform future iterations of introductory computer science curricula, demonstrating: (1) engagement of students in solving problems using quantum computing concepts, (2) practical strategies for incorporating introductory quantum computing modules into computer science curricula, and (3) an exploratory framework for computing education that prepares technically proficient, socially, ethically responsible innovators for a post-quantum future.

Designing with AI: Using Generative Tools to Build and Revise Courses*

Work in Progress

Stephanie Hoeppner
Assistant Professor, Information Technology
University of Cincinnati Clermont, Batavia, OH 45103
hoeppnse@ucmail.uc.edu

Krist Hall
Associate Professor, Information Technology
University of Cincinnati Clermont, Batavia, OH 45103
hall3kj@ucmail.uc.edu

Faculty in the Information Technology program at the University of Cincinnati Clermont College are using generative AI to develop a new "Survey of Information Science and Technology for Healthcare Professionals" course and update our existing "Fundamentals of Information Technology" course. Our focus is on leveraging AI to ensure alignment between course objectives, program outcomes, and current industry standards while streamlining development.

Our iterative approach involves crafting targeted prompts to generate course content, then using AI output to create follow-up prompts that refine materials. For the healthcare survey course, we used AI to generate module-level outcomes aligned with course-level outcomes, ensuring assignments, assessments, and materials aligned with module outcomes. For the fundamentals course update, AI identified gaps between existing curriculum and current IT trends, generating updated content areas tied to learning objectives and corresponding instructional materials.

Key insights from our work include: First, using multiple AI tools provides diverse perspectives and richer results. Second, AI excels at creating objective grading rubrics that assess assignments consistently. Third, AI tools effectively help faculty ensure comprehensive alignment throughout courses, confirming that teaching and assessment truly match delineated outcomes. However, success depends on learning to craft appropriate prompts for desired results. We also learned that effective AI collaboration requires domain expertise to evaluate and refine generated content, particularly for technical accuracy and industry relevance.

This presentation will share specific examples of our prompt engineering strategies, demonstrate actual AI-generated course materials, and discuss critical lessons learned about maintaining academic rigor while leveraging AI efficiency. Our goal is to provide concrete, actionable insights for faculty considering AI-assisted curriculum development in technical fields.

Allowing Late Course Work With No Grade Penalty*

Work in Progress

David L. Largent
Department of Computer Science
Ball State University, Muncie, IN 47306
dllargent@bsu.edu

I have been using specifications grading since 2017. In addition, I have incorporated many learner-centered aspects into my courses, such as allowing learners to help establish grading policies. For many years, I have gradually drifted towards making my approach to grading as equitable as possible, attempting to remove all possible barriers to success as possible, and evaluating learners based on their demonstrated knowledge of the course material and not on their performance, such as meeting deadlines. During the last academic year, I started accepting (re)submissions of course work late, with no grade penalty. I still set due dates, but learners could submit their work anytime during the semester, up to the end of the semester, and still earn full credit for it, if correct. I will briefly report on my experiences and what changes I have made for the courses this fall.

Ethical Issues in Computer Science and AI*

Conference Tutorial

Harris, Andrew J
Sr. Lecturer of Computer Science
Ball State University

The goal of this discussion is to consider the ethical considerations that have always been a part of computer science education. While there has long been a need to consider ethical conduct in computing (https://www.acm.org/code-of-ethics). It is clear that AI is bringing even more uncertainty, and it is critical that we address these issues in addition to the more technical material we learned in our training. We will start with the following questions, but mainly we want a chance to think through some of these issues together.

1. How do we integrate ethics instruction (for example ACM guidelines) into our curricula? Is it a stated goal? If not, how do we ensure students have some knowledge of these topics?
2. Do we have an obligation to teach enough computer science principles to non-majors that they have at least an informed view of some of the concerns wrought by new technologies (especially AI)?
3. How do we address the role technologies are playing in some of the hot button issues of our time, including (but not limited to):
   (a) Privacy issues
   (b) Gatekeeping
   (c) Intellectual property concerns
   (d) Freedom of speech
   (e) Information (news) siloing
4. How do we begin to cope with AI? Of course we want to teach the science of it, but the ethical issues this technology introduces are mind-numbing.

It is conceivable we will have code written by AI being graded by AI. Is that the future we want? How do we encourage responsible use of AI and related technologies?

Discussion on Needed Changes to the Curriculum Because of AI*

Conference Tutorial

Cathy Bareiss
Bethel University
Mishawaka, IN 46545
cbareiss@acm.org

This is a round table discussion as we all look to adapt our curriculum to deal with the changes AI has brought. What should we teach out computing students and at what level? What courses need to be adapted, added, or removed? While the ACM CS2023 curriculum guidelines do address the need of AI, the changes over the past two years brought on by AI requires us to adapt our curricula even more. Come to the round table and share your thoughts and learn from others as we discuss this issue.

Course In A Box: Computing History and Ethics*

Conference Tutorial

James K. Huggins
Department of Computer Science
Kettering University
Flint, MI 48504
jhuggins@kettering.edu

In this tutorial, we present a "Course In A Box" for a course in computing history and ethics. This course has been taught at Kettering University for the last twenty years, evolving from an elective course offered periodically to a required course offered every term. The presenter is the original course designer and primary instructor for the course. Features of the course include:

- Replacement of lectures during the first half of the course by classroom discussions of required readings, combining the "flipped classroom" and "one-minute paper" pedagogical techniques [4]. (Thus, the course can be taught without the use of slide decks.)
- Replacement of lectures during the second half of the course by student oral presentations on self-selected topics in computing history and computing ethics. The presentations serve as an informal draft of the formal papers submitted later.
- Supplemental readings with frameworks for historical & ethical analysis:
  - A paper on value propositions using an NABC ("need", "approach", "benefit", "competition") approach to describing the significance of a historical artifact [1].
  - A paper presenting an "algorithmic" (i.e. systematic) approach to ethical analysis [2].

- Emphasis on real-world ethical problems rather than hypotheticals [3].
- Supplemental video presentations as "filler" when course pauses are needed.

References

[1] C.R. Carlson and W.W. Wilmot. *Innovation: The Five Disciplines for Creating what Customers Want*. Business book summary. Crown Business, 2006.

[2] W.Robert Collins and Keith W. Miller. Paramedic ethics for computer professionals. *Journal of Systems and Software*, 17(1):23–38, 1992. Computer Ethics.

[3] James K. Huggins. Computing history & ethics: the relevance of the real world for social change. *SIGCAS Comput. Soc.*, 45(2):34, July 2015.

[4] James K. Huggins. Flipping unplugged: An experience report. In *2018 IEEE Frontiers in Education Conference (FIE)*, pages 1–4, 2018.

Vibe Coding*

Conference Tutorial

Harris, Andrew J
Sr. Lecturer of Computer Science
Ball State University

1 Introduction

Vibe coding is the practice of generating code using AI tools. The appeal of this approach is that applications can be generated with very little programming knowledge. The advent of this idea has already had a number of interesting implications: If a non-coder can build an app with AI, why continue to train coders? This technique could lead to massive efficiencies It democratizes app development, turning anybody into a developer It may lead to fewer jobs in software development

In practice, it has turned out to be a very mixed bag. While AI tools can generate code, it doesn't always run correctly The coding style may be weak or inconsistent Such code is rarely built to scale well There can be significant security issues in the resulting code The code can be extremely difficult to maintain Students don't learn coding fundamentals if they don't produce code.

2 Plan

In this session, participants will practice vibe coding to see for themselves what this is all about. Please bring a laptop and access to some of your assignments. We'll try a few techniques to see how students may be using this technology, how to detect its use, and go over a few techniques where the practice may be helpful or useful. We will focus on freely available AI models, but if you have access to a higher-end AI tool, you are welcome to use it in this exercise.

Possible Outline:

- Coding with chatGPT
- Vague prompts
- Testing your assignment instruction (AI bombs)
- Using AI for pseudocode development
- Iterative development
- AI for asset management

State Machines: Tools, Conversions and Implementations*

Nifty Assignment

Ramachandra B. Abhyankar
Indiana State University
RB.Abhyankar@indstate.edu

Use of state machines has been very successful and widespread in hardware design. Attempts are being made, to make state machines the basis for software development as well. This makes state machines extremely important in Computer Science and Engineering. An in-depth knowledge of state machines is now very important for Computer Science and Engineering students. This assignment deepens the understanding of state machines. At Indiana State University, in Terre Haute, Indiana, this assignment is part of the Computer Organization course which covers both Computer Hardware and Computer Software topics. Statement of the assignment: Develop Mealy and Moore Abstract State Machines to solve the "multiply by 2" problem for unsigned binary integers. Test if your state machines are correct by testing the machines using the JFLAP tool . Code the Mealy and Moore State Machine in the C programming language and test it. Finally, implement the State Machine in hardware (using flip-flops and logic gates). Simulate using the LogiSim tool. Write a report describing all your work, and the results of testing your work. Turn in your report. The presentation at the conference will describe the solution in detail. This assignment provides an in-depth exploration of both Mealy and Moore state machines and conversions between them. The assignment gives students a chance to become familiar with the use of the JFLAP tool and the LogiSim tool. This assignment covers implementation of state machines in both hardware (using flip-flops and logic gates) and software (as C programs). This assignment helps to bring together many different topics covered in a Computer Organization course.

Sparse Matrix-Matrix Multiplication with CUDA and Python Integration*

Alina Lazar
Computer Science and Information Systems
Youngstown State University
Youngstown, OH 44555
alazar@ysu.edu

As parallel computing becomes increasingly relevant across various domains, from scientific simulations to machine learning, the development and accessibility of multi-core and GPU-enabled hardware platforms have made high-performance computing more widespread. This shift presents opportunities and challenges for computer science educators, especially those teaching parallel and distributed computing (PDC). This assignment focuses on implementing sparse matrix-dense matrix multiplication (SpMM) using CUDA and integrating it into Python with CUDAExtension, providing students with hands-on experience optimizing computational workloads.

Sparse matrix multiplication [1] arises in numerous applications, including computational physics, graph analytics, machine learning, and large-scale simulations. Traditional CPU-based implementations of sparse matrix multiplication often suffer from performance bottlenecks due to memory access patterns and limited parallelization capabilities. GPUs offer a promising solution for accelerating these operations. Unlike dense matrices, where most elements are nonzero, sparse matrices contain many zero elements, making efficient storage and computation essential.

The proposed assignment involves implementation and performance analysis [2], teaching students to evaluate and optimize the efficiency of their code. The assignment encourages students to innovate and experiment with different approaches to implementing the SpMM, fostering creativity and critical thinking skills that are highly valued in technology fields.

Acknowledgments

This research was supported by the U.S. Department of Energy's Office of Science, Office of High Energy Physics, of the US Department of Energy under Contracts No. DE-SC0024364 (FAIR).

References

[1] Guyue Huang et al. "Ge-spmm: General-purpose sparse matrix-matrix multiplication on gpus for graph neural networks". In: *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE. 2020, pp. 1–12.

[2] Donghyeon Kim and Jinsung Kim. "Analysis of Several Sparse Formats for Matrices used in Sparse-Matrix Dense-Matrix Multiplication for Machine Learning on GPUs". In: *2022 13th International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE. 2022, pp. 629–631.

Gamification of a Game Development Course*

Nifty Assignment

Forrest Stonedahl
Mathematics and Computer Science Department
Augustana College, Rock Island, IL 61201
forreststonedahl@augustana.edu

In this work, I share my experience using a gamified grading system in a project-based upper-level CS course on video game development to incentivize students to engage in a combination of competency-based and community-based learning. While the student audience was presumably self-selected to have an affinity for games and the course topic made the concept of a "playing a game" to earn grades thematically appropriate, some of the gamification mechanisms could transfer effectively to other courses/contexts.

A system of badges drew on ideas from specifications grading [1] wherein students' "player characters" earn badges for demonstrating specific technical skills and different numbers of completed badges were required to "level up" their character and unlock the ability to earn higher grades. A key goal was to motivate students to help each other learn more and build better projects by fostering a learning community through peer-instruction. One simple approach awarded "sharing stars" to students for class-wide presentations of useful tips they had discovered. A more complex element involved "Scrolls of Aid" which students could earn (or purchase with their character's in-game gold), and then offer to another student in exchange for help with some aspect of their project. The assisting student would then earn an "Artifact of Aid" by documenting the assistance rendered. This approach motivated students to both help and be helped by other students in the class. Overall, the gamified system worked well, although there is room for refining incentives and streamlining the bookkeeping.

References

[1] Linda B. Nilson. *Specifications grading: Restoring rigor, motivating students, and saving faculty time.* Routledge, 2015.

Exploring the Strengths and Limitations of Word2Vec through Analogy Questions*

Nifty Assignment

Hong Biao Zeng
Department of Computer Sciences
Fort Hays State University
Hays, KS 67601
h_zeng@fhsu.edu

This can serve as the first programming, discussion, and/or research assignment for a class of Introduction to Natural Language Processing. Students will explore the strengths and limitations of Word2Vec embeddings by solving analogy problems that are similar to those found on standardized tests like GRE.

The students are asked to implement a Python function that uses a pretrained Word2Vec model to answer analogy questions. Given a question word pair and five candidate word pairs as multiple-choice options, the function should return the pair that best completes the analogy.

A sample input of the function could be:

- Question pair: ("walk", "legs")
- Choices: [("gleam", "eyes"), ("chew", "mouth"), ("smile", "face"), ("cover", "book"), ("grind", "nose")].

The expected output is one of the choice pairs that best matches the analogy relationship in the question pair.

While the students may be excited to see their functions correctly solve a few problems, they may also be disappointed to find that their functions perform poorly on many others. The students are encouraged to further research the limitations of distributional semantics and examine the challenges of modeling deeper semantic relationships using Word2Vec alone. The students' work will be evaluated based on both code implementation and the discussion of the limitations of Word2Vec embeddings.

Since this is the first assignment of the entry-level course of NLP, the students may be provided with the skeleton code or pseudo-code that outlines the approach.

Effective Strategies, Challenges, and Best Practices in Teaching Cybersecurity to College Students: A Faculty-Based Analysis *

Imad Al Saeed
Computer Sciences Department
Saint Xavier University
Chicago, IL 60655
alsaeed@sxu.edu
773-298-3393

Abstract

As cybersecurity threats proliferate in our increasingly digital world, higher education faces the dual challenge of seizing the timely opportunity to train adept professionals and overcoming significant instructional barriers. This study synthesizes faculty insights and existing scholarships to present a unified framework for strengthening cybersecurity education at the college level. Drawing on qualitative responses from experienced U.S. professors, in-depth case studies, and educational theory, we identify hands-on, experiential learning as the cornerstone of effective instruction, highlight the value of interdisciplinary integration, particularly ethics, psychology, and real-world scenarios for deepening student engagement, and underscore the necessity of up-to-date industry-aligned curricula. We further examine key institutional obstacles, including resource constraints, uneven faculty development support, and curriculum inertia, and offer best practices for cultivating faculty expertise, fostering collaboration across departments, and leveraging active-learning technologies. Together, these findings provide a comprehensive, practice-oriented

roadmap for preparing college students to meet the evolving demands of the cybersecurity field.

1 Introduction

Cyberneticist has emerged as one of the fastest-growing disciplines in technology and education, driven by the increasing frequency and sophistication of digital threats and the rapid evolution of information technology. Colleges and universities therefore bear a critical responsibility to equip students with not only the technical expertise to secure systems, networks, and data but also an understanding of the ethical, legal, and societal implications of their work [4][19]. In recent years, cybercrime has grown into multibillion dollar industry. Data breaches, ransomware attacks, and digital espionage now regularly dominate headlines underscoring the urgency of adaptive and responsive curricula that prepare students to defend against such threats [14]. However, the dynamic nature of cybersecurity poses significant challenges to traditional teaching approaches. Instructors must strike a balance between technical rigor and interdisciplinary context, integrating ethics, psychology, and real-world scenarios into hands-on, experiential learning [14][12]. At the same time, institutions must overcome resource constraints, curriculum inertia, and uneven faculty development support to foster rapid curricular evolution and a culture of security awareness across all fields of study [16]. This paper investigates the pedagogical approaches, curriculum design, and instructional challenges involved in teaching cybersecurity to college students. Drawing on qualitative data from experienced faculty across various United States universities such as University of Texas, Cal Poly Pomona, Florida State University, Arizona State University, University of Michigan, Howard University, University of Colorado, University of Washington, Georgia Tech, and Rutgers University, existing literature, and case studies, it aims to analyze effective strategies, highlight recurring barriers, and propose best practices for cultivating not only technical competency but also critical thinking and ethical awareness and ultimately providing educators with a holistic framework for training the next generation of cybersecurity professionals.

2 Literature Review

Prior research highlights a growing demand for robust cybersecurity education models that keep pace with industry developments. Bishop (2018) emphasizes the need for pedagogical reform to match the complexity of evolving cyber threats [9]. Freeman et al. (2014) argue for active learning approaches that increase student retention and performance in technical disciplines [3]. Kolb's Experiential Learning Theory (1984) supports the integration of hands-on activities to bridge theory and practice [13]. Rotenberg (2016) and Hadnagy (2018) underscore the importance of ethical reasoning and understanding human behavior in developing holistic cybersecurity professionals [20][11]. Meanwhile, Green and Smith (2019) document the strain placed on faculty by insufficient institutional support, calling attention to burnout and resource limitations [10]. This study builds on this foundation by providing updated qualitative insights directly from instructors currently engaged in cybersecurity education across the United States.

2.1 The Importance of Cybersecurity Education

Cybersecurity is no longer a niche area, it is a cornerstone of national security, corporate integrity, and personal privacy. The global shortage of cybersecurity professionals has further emphasized the need to produce graduates who are not only technically competent but also capable of ethical decision-making and critical analysis. According to the (ISC)^2 Cybersecurity Workforce Study (2023), the global cybersecurity workforce needs to grow by 65 percent to meet current demand [11]. Moreover, cyberattacks are increasingly targeting educational institutions themselves, highlighting the urgency of cybersecurity awareness among students [14]. Cybersecurity education can empower students to contribute to protecting digital infrastructure while also preparing them for lucrative and meaningful careers. With the growing integration of digital tools in all fields, even non-computer science majors benefit from basic cybersecurity literacy [4]. In fact, integrating cybersecurity fundamentals into general education courses can promote a culture of digital responsibility and awareness [19].

2.2 Curriculum Development and Core Competencies

Designing an effective cybersecurity curriculum involves aligning educational objectives with industry standards, professional competencies, and emerging trends. Key competencies should include:

- Network and System Security
- Cryptography and Data Protection
- Risk Management and Governance
- Penetration Testing and Ethical Hacking
- Incident Response and Digital Forensics
- Security Policy, Compliance, and Legal Considerations

Frameworks such as the National Initiative for Cybersecurity Education (NICE) provide guidelines to ensure educational programs are aligned with workforce needs [8]. Furthermore, curriculum design must be adaptive to include new content as threats evolve and technologies change. Interdisciplinary integration with courses in law, psychology, and ethics enriches the program and better prepares students for real-world scenarios [18]. Additionally, including modules on emerging technologies such as blockchain security, artificial intelligence, and quantum cryptography can ensure the curriculum remains forward-looking. Institutions should also consider stackable credential programs that allow students to gain industry-recognized certifications alongside their academic degrees [17].

2.3 Teaching Strategies for Cybersecurity Education

Effective cybersecurity education requires a blend of theoretical instruction and practical application. Several instructional strategies have proven effective:

- Hands-On Learning Virtual labs and simulation environments enable students to practice their skills in a controlled setting. Cyber ranges and online platforms like Cybersecurity Labs, NetLab, or TryHackMe allow students to experience real-world scenarios safely [6].
- Case Studies and Scenario Analysis Analyzing past cyber incidents (e.g., the Equifax breach, SolarWinds attack) helps students understand the practical consequences of security failures. This also enhances critical thinking and ethical reasoning.
- Problem-Based Learning (PBL) PBL fosters collaborative learning and problem-solving skills. Students work in teams to address open-ended problems, often simulating the multidisciplinary nature of cybersecurity work [4].
- Guest Lectures and Industry Engagement Bringing industry professionals into the classroom bridges the gap between academic learning and real-world practice. Industry partnerships can also provide students with internship opportunities and mentorship.
- Gamification and Competitions Capture the Flag (CTF) competitions and gamified environments make learning engaging while fostering practical skills in cryptography, reverse engineering, and network defense [6].
- Online Learning and Hybrid Instruction Given the rise of remote education, incorporating online modules, interactive tools, and asynchronous content delivery can enhance accessibility and flexibility for diverse learners [1].

2.4 Challenges in Teaching Cybersecurity

Despite the growing importance of cybersecurity education, several challenges persist:

- Rapid Technological Evolution The fast-paced nature of cybersecurity means course content must be constantly updated. This requires significant effort from faculty and support from institutions [18].
- Limited Qualified Instructors There is a shortage of educators who possess both academic credentials and real-world cybersecurity experience. Institutions must invest in faculty development and industry collaboration.
- Students often have diverse levels of technical knowledge. This can complicate instruction and necessitate differentiated teaching approaches [14].
- Ethical and Legal Concerns Teaching skills such as penetration testing requires a strong ethical foundation and clear communication about legal boundaries to prevent misuse [4].
- Resource Constraints Cybersecurity labs and simulation environments require substantial technological and financial resources. Not all institutions can afford these tools, potentially widening the gap in educational quality [19].

2.5 Assessment and Evaluation Methods

Assessing student learning in cybersecurity should go beyond multiple-choice tests. Effective assessment strategies include:

- Lab Practicals: Allow students to demonstrate their technical proficiency.
- Capstone Projects: Offer comprehensive assessments where students design secure systems or perform security audits.
- Peer Review and Presentations: Foster communication and critical analysis.
- Reflective Essays: Encourage students to consider the ethical and social implications of cybersecurity.

Formative and summative assessments should be aligned with course learning objectives and industry expectations [8]. Tools like cybersecurity competency rubrics and performance-based assessments can further enhance evaluation [1].

2.6 Best Practices and Recommendations

To strengthen cybersecurity education, institutions and instructors should consider the following recommendations:

1. Integrate Interdisciplinary Perspectives: Include topics from ethics, law, and psychology to provide broader context.
2. Support Faculty Development: Encourage faculty to pursue certifications (e.g., CISSP, CEH) and engage in ongoing training.
3. Foster Industry Collaboration: Partner with cybersecurity firms to provide internships, guest lectures, and curriculum input.
4. Use Adaptive Learning Technologies: Tools that adjust to students' proficiency levels can help manage diverse classrooms.
5. Promote Diversity and Inclusion: Recruit and support underrepresented groups in cybersecurity to address workforce gaps.
6. Emphasize Soft Skills: Communication, collaboration, and critical thinking are essential in multidisciplinary cybersecurity teams.
7. Encourage Research and Innovation: Provide opportunities for students to engage in cybersecurity research and development, fostering innovation and entrepreneurial thinking.

3 Research Questions

1. What teaching strategies are most effective in cybersecurity education?
2. How do interdisciplinary approaches contribute to student learning outcomes?
3. What institutional challenges hinder the delivery of the current cybersecurity curricula?

4 Hypothesis

1. Hands-on, experiential learning significantly improves student engagement and understanding in cybersecurity education.
2. The interdisciplinary integration of ethics, law, and psychology enhances cybersecurity students' critical thinking and professional preparedness.
3. Lack of instructor preparedness and institutional support are major barriers to implementing up-to-date cybersecurity curricula.

5 Methodology

This study employs a qualitative research design to explore instructional practices in cybersecurity education. Fifteen professors teaching undergraduate and graduate cybersecurity courses at various U.S. universities were purposively selected based on their teaching experience and disciplinary diversity. Participants provided written interviews and survey responses, which were then anonymized to encourage candor and protect confidentiality [15]. Responses were systematically coded and organized into three a priori themes teaching strategies, interdisciplinary integration, and institutional challenges and further refined through iterative thematic analysis following Braun and Clarke's six-phase framework (2006) [5]. To enhance rigor and validity, methodological triangulation was applied by comparing findings across multiple institutions and respondent backgrounds [7]. Data triangulation helped confirm recurring patterns and guard against single-source bias. Overall, this combined approach purposive sampling of experienced faculty, anonymized written data collection, thematic coding, and triangulation provides a robust foundation for identifying effective pedagogical practices and barriers in higher-education cybersecurity programs.

6 Results and Analysis

These Faculty statements such as "Students who worked on real-time penetration testing in our cyber range scored 25 percent higher in final assessments and showed more confidence in group interviews.", "Our switch to project-based learning increased participation, especially among underrepresented students. Virtual labs made concepts click in a way lectures never could.", and "Students often cite the practical activities, like mock incident responses, as the most valuable parts of the course." confirm the hypothesis that hands-on learning is essential in cybersecurity education. Real-world simulations, lab environments, and project-based learning were identified as major contributors to student understanding and retention. These approaches align with active learning theories [9] and support the development of practical skills crucial for professional success. Experiential learning promotes not only engagement but long-term knowledge retention. According to Kolb's Experiential Learning Theory (1984), students learn best through direct experience and reflection, both of which are present in cybersecurity labs and simulations [13]. The other faculty responses such as "After adding modules on legal frameworks and human factors, students asked deeper questions and showed a more holistic understanding of cyber incidents.", "Students better appreciated the 'why' behind cybersecurity protocols after exploring psychological manipulation techniques in phishing.”, and “Ethics case studies sparked intense classroom debates and led to better performance on open-ended assessment tasks.” reinforce the importance of interdisciplinary teaching in cybersecurity programs. Incorporating ethical dilemmas, legal implications, and psychological tactics not only broadens students’ perspectives but also prepares them for real-world challenges. Interdisciplinary learning fosters critical thinking, ethical responsibility, and effective decision-making [21]. Incorporating modules on cyber law and digital ethics has been shown to deepen students’ comprehension of privacy, surveillance, and digital rights [20]. Furthermore, cybersecurity threats often involve manipulating human behavior, making psychological literacy essential [11]. The final faculty responses such as “Over 60 percent of faculty feel underprepared to teach emerging threats like AI-driven attacks due to limited institutional training.”, “We struggle to retain cybersecurity faculty because industry jobs offer much higher pay.”, “Without funding for new lab tools, our students can’t experience modern techniques like blockchain forensics or cloud security first-hand.”, and “Institutional investment in faculty certifications like CISSP has a direct impact on teaching quality, yet many schools don’t offer reimbursement or time off.” confirmed the faculty support and resource allocation were critical issues. Without institutional backing, educators face difficulties updating curricula, acquiring new skills, or competing with industry salaries. These barriers can lead to outdated coursework and inadequate student preparation. Supporting faculty through ongoing training and funding is essential for sustaining a robust cybersecurity education [2]. The National Initiative for Cybersecurity Education [8] highlights professional development and continuous learning as pillars of sustainable cybersecurity instruction. Faculty burnout and turnover also reduce program quality [10].

7 Discussion

The faculty insights offer consistent support for each hypothesis. Hands-on learning remains the cornerstone of effective cybersecurity education. Interdisciplinary modules, while still developing in many programs, are proving their value in equipping students with broader perspectives. Institutional issues, though more difficult to resolve, represent a major barrier to educational quality. These findings indicate the need for national and institutional-level reforms. Programs designated as NSA Centers of Academic Excellence have begun addressing these gaps, but broader adoption is required. Enhancing faculty retention and updating lab tools will be critical to preparing students for complex cybersecurity careers.

8 Conclusion

This research identifies three core elements of successful cybersecurity education such as experiential learning, interdisciplinary integration, and strong institutional support where each of which is essential for cultivating a workforce capable of responding to evolving digital threats. Faculty testimonies confirm that applied hands-on instruction and the inclusion of ethics and psychology significantly enhance student learning and engagement. However, systemic barriers such as underprepared faculty and insufficient institutional investment continue to hinder the rapid evolution of cybersecurity curricula across higher education. Teaching cybersecurity to college students thus remains a multifaceted and dynamic challenge, demanding innovative instructional strategies, interdisciplinary curricula, and a steadfast commitment to ethical education. By integrating hands-on learning, real-world applications, and continuous adaptation, higher education institutions can ensure graduates are not only technically competent but also confident in confronting the cybersecurity challenges of tomorrow.

9 Recommendations

- Implement lab-based and project-driven courses that simulate real-world cyber threats.
- Integrate ethics, legal frameworks, and behavioral science into core cybersecurity curricula.
- Develop faculty certification and sabbatical programs to enhance instructor preparedness.
- Modernize lab infrastructure and tools to reflect current and emerging cyber threats.
- Establish and strengthen industry-academic partnerships for curricular relevance and student placement.

10 Future Directions

1. As cybersecurity threats become more integrated with emerging technologies such as AI, IoT, and quantum computing, future curricula must evolve to address these domains. Research into effective pedagogical models, especially online and hybrid formats, is also crucial. Educational institutions must prioritize cybersecurity not only in computer science departments but across all disciplines.

2. Moreover, global collaboration and standardization in cybersecurity education could help ensure consistency and scalability of programs. Initiatives like the Global Forum on Cyber Expertise (GFCE) and international partnerships between universities can foster cross-border educational innovation.

References

[1] M Alshaikh. "Effective cybersecurity education: Strategies for improving learner engagement in online environments". In: *European Conference on Games Based Learning*. Vol.1. 2021, pp. 1–15.

[2] R. Anderson et al. "Measuring the cost of cybercrime". In: *Springer*. 2021.

[3] M. Bishop. "Education in cybersecurity." In: *Communications of the ACM*. 61(6). 2018, pp. 44–47.

[4] M. Bishop and D. Frincke. "Teaching practical cybersecurity". In: *IEEE Security Privacy*. 4(6). 2006, pp. 54–56.

[5] V. Braun and V. Clarke. "Using thematic analysis in psychology." In: *Qualitative Research in Psychology*. 3(2). 2006, pp. 77–101.

[6] A. Conklin et al. "Principles of Computer Security (4th ed.)" In: *McGraw-Hill Education*. 2018.

[7] J. W. Creswell and C. N. Poth. "Qualitative inquiry and research design: Choosing among five approaches." In: *Sage publications*. 2017.

[8] National Initiative for Cybersecurity Education (NICE). "Cybersecurity workforce framework (NIST Special Publication 800-181 Rev. 1). Retrieved from https://www.nist.gov/itl/applied-cybersecurity/nice". In: 2022.

[9] S. Freeman et al. "Active learning increases student performance in science, engineering, and mathematics." In: *Proceedings of the National Academy of Sciences*. 111(23). 2014, pp. 8410–8415.

[10] B. Green and J. Smith. "Faculty burnout and retention in cybersecurity education." In: *Journal of Higher Education Policy and Management*. 41(3). 2019, pp. 310–322.

[11] C. Hadnagy. "Social engineering: The science of human hacking. Wiley." In: 2018.

[12] C. Irvine and M. Thompson. "Challenges in cybersecurity education." In: *IEEE Security Privacy*. 18(2). 2020, pp. 78–81.

[13] D. A. Kolb. "Experiential learning: Experience as the source of learning and development." In: Prentice Hall. 1984.

[14] N. Kshetri. "Cybersecurity in Higher Education: Challenges and Solutions. IT Professional." In: 23(2). 2021, pp. 63–67.

[15] S. Kvale and S. Brinkmann. "InterViews: Learning the craft of qualitative research interviewing (2nd ed.)." In: 2009. Los Angeles, CA: Sage Publications.

[16] Mandiant. "Cyber security threats and trends. Retrieved from https://www.man In: 2022.

[17] D. Manson and F. Curling. "Cybersecurity education and workforce development: Academic programs and alternative pathways." In: *Information Systems Education Journal.* 16(5). 2018, pp. 4–10.

[18] C. P. Pfleeger and S. L. Pfleeger. "Security in Computing (5th ed.). Prentice Hall." In: 2011.

[19] P. Pusey and W. Sadera. "Building cybersecurity programs in higher education. Retrieved from https://csiac.org/articles/building-cybersecurity-programs-in-higher-education/". In: Cybersecurity and Information Systems Information Analysis Center (CSIAC). 2020.

[20] M. Rotenberg. "Privacy and cybersecurity: The new frontier. Harvard Journal of Law Public Policy, 39, 493." In: 2016.

[21] R. Sharma and A. Sefcheck. "Ethical reasoning and cybersecurity education." In: *Journal of Ethics and Information Technology.* 21(2). 2019, pp. 113–123.

From Conditionals to Consequences: A Framework for Integrating Ethics Labs Across the CS Curriculum*

Heather Amthauer[1] and Peter Ohmann[1]
[1]College of Saint Benedict and Saint John's University
hamthauer001@csbsju.edu
pohmann001@csbsju.edu

Abstract

This position paper proposes a structured framework for embedding ethical reasoning throughout the early computer science curriculum using dedicated ethics labs. Rather than reserving ethics for isolated upper-division courses or one-off modules, we advocate integrating ethics into core technical classes from CS0 to CS2. These labs combine coding practice with real-world dilemmas, such as algorithmic bias, automated decision-making, and structural inequality, to prompt ethical reflection alongside skill development. Grounded in student development theory and the concept of ethical fluency, our framework aligns ethical reasoning with technical progression, helping students form their professional identity while they learn to code. We describe four example labs, each aligned with a different course level, and show how these experiences scaffold into upper-division electives. By making ethics an early and recurring part of technical learning, this framework helps students see that responsible computing begins in year one and continues throughout their academic journey.

1 Introduction

This position paper argues for the systematic integration of ethical reasoning into the early computer science curriculum through the use of dedicated, scenario-based ethics labs. As computing systems play increasingly influential roles in society, ethical reasoning must become a foundational competency for computer science students, not an afterthought introduced only in upper-division electives. Yet in many programs, ethics remains isolated, covered in a single standalone course or addressed through brief, disconnected assignments. These approaches, while well-intentioned, are insufficient for preparing students to engage in the kind of reflective, value-sensitive decision-making that real-world software development demands.

We propose a more immersive and iterative model: a curricular framework that embeds structured ethics labs across the early CS sequence, beginning in CS0 or CS0.5 and extending through CS2. These labs are tightly integrated with core technical material, such as conditionals, object-oriented programming, and data structures, and provide hands-on opportunities for students to grapple with ethical dilemmas such as algorithmic bias, content moderation, and autonomous decision-making. Rather than treating ethics as supplemental, this framework treats ethical reasoning as an essential part of what it means to write code responsibly.

Throughout this paper, we use the term *lab* to refer to structured, instructor-guided sessions, typically held weekly or semi-weekly, where students actively write and test code in response to real-world scenarios. Labs are conducted in small groups or pairs and are distinct from lectures or homework in that they emphasize real-time experimentation, peer collaboration, and immediate support. This format is especially well-suited for ethical exploration: it creates space for students to not only build functioning programs but also to reflect, together, on their broader implications.

We present this lab framework not merely as a pedagogical tool for a single assignment but as a curricular structure that scaffolds ethical reflection across multiple technical courses. Each lab is aligned with the concepts being taught at its respective level while laying the groundwork for continued ethical engagement in upper-division courses like AI, bioinformatics, software engineering, and computing ethics seminars. By mapping ethical reflection to technical progression, we aim to show that ethics can and should be treated as integral to computing education from the very beginning.

2 Background and Foundations

Ethics labs are especially powerful when introduced in the first two years of a computer science major, during the same period students are building foundational programming skills and shaping their professional identity. Chickering's theory of identity development emphasizes the formation of competence, purpose, and integrity during early college years [5, 7]. Similarly, Carlone and Johnson's Science Identity Framework [4] argues that recognition, being seen by others and oneself as a legitimate participant in STEM, is a crucial part of identity development, particularly for students from underrepresented groups. Ethics labs support these dimensions by giving students opportunities to demonstrate competence, engage in realistic coding practices, and receive recognition as thoughtful, responsible technologists.

In addition, Astin's Involvement Theory [1] emphasizes the link between meaningful academic engagement and student success. Ethics labs foster this kind of engagement by asking students to confront real-world dilemmas, such as algorithmic bias or decision-making under uncertainty, through hands-on coding and structured reflection. Branch and George [2] further highlight that reflection, particularly when it follows moments of uncertainty or challenge, contributes directly to moral and professional development. When students write code that could benefit or disadvantage individuals based on how a function is implemented, the stakes of their technical work become visible and worth discussing.

Beyond supporting student identity and involvement, these early experiences also build what Landon and Schultz [11] describe as ethical fluency: the ability to recognize, interpret, and thoughtfully respond to ethical dilemmas in real time. Ethical fluency is not something students develop by learning principles alone, it requires practice. Repeated exposure to moral complexity within the context of technical tasks helps students build the habits of mind required to reason through ambiguity, revise decisions, and engage with competing perspectives.

Finally, the lab environment itself supports this developmental work. Labs provide a structured but flexible space that embodies what Sanford [12] describes as the necessary balance of challenge and support for student growth. When students encounter ethical ambiguity in a lab, such as deciding whom a self-driving car should prioritize, they are not left to process that tension alone. Instead, they navigate it in collaboration with their peers and with support from an instructor, often revising their thinking in real time. This scaffolding allows students to take intellectual risks while also giving them the guidance to grow from those experiences.

Together, these developmental, cognitive, and pedagogical perspectives all point toward a shared conclusion: early, lab-based experiences are uniquely effective at integrating ethical reflection into students' technical identity. They not only help students learn to code, but also to understand the impact of that code, and to care about what it does in the world.

3 Related Work and Curricular Models

In many computer science programs, ethics is confined to a single standalone course, often delivered in the upper division and included to meet accreditation requirements. Fiesler et al. [8] note that this structure can unintentionally signal that ethical reasoning is peripheral, something to be addressed only after students have mastered the "real" technical material. This separation reinforces the false notion that technical work and ethical reflection are distinct, when in practice, they are deeply intertwined.

In response to the limitations of siloed courses, educators have explored a variety of embedded ethics models that introduce ethical reasoning within technical coursework. These approaches aim to make ethics feel more relevant by integrating it into the day-to-day experience of learning to code.

One common method is micro-insertion: brief, targeted prompts or scenarios embedded into technical lessons. These "bite-sized" interventions require minimal curricular change and can boost ethical awareness. For example, Brown et al. [3] found that students exposed to micro-insertions showed increased sensitivity to the ethical dimensions of technical decisions.

Other models take a narrative approach. The Computing Ethics Narratives (CEN) project [6], offers a repository of real-world media, from podcasts to articles, that prompt discussion and ethical reflection through storytelling. Similarly, Embedded EthiCS integrates short modules directly into technical coursework, pairing CS instructors with philosophers to guide ethical analysis of justice, bias, and responsibility [9].

Some instructors use contextualized programming assignments that embed ethical dilemmas into code implementation tasks. Fiesler et al. [8], for example, developed ethics-infused projects for introductory programming courses. These approaches encourage students to reason through trade-offs as part of the coding process.

While these strategies enrich the curriculum, they are often brief, discussion-based, or loosely connected to the programming work itself. Students may reflect on ethics conceptually but rarely test or revise their values through hands-on iteration. Dedicated ethics labs address this gap. By embedding ethical reflection into the full coding lifecycle, from implementation to debugging to discussion, labs make ethical reasoning part of technical practice. Labs offer students the space to build, test, and revise ethically charged decisions in a collaborative, instructor-supported setting. They are not simply about covering ethics content. They are about practicing ethical reasoning as part of what it means to code.

4 A Scaffolded Framework for Ethics Integration

Rather than treating ethics as a separate or occasional component of the computer science curriculum, we propose a scaffolded framework that integrates structured ethical reflection into core technical courses—beginning as early as CS0 and continuing through CS1, CS2, and beyond (see Figure 1). Each ethics lab is embedded within a course where it reinforces the technical content students are learning, while simultaneously introducing them to the ethical dilemmas they may face as future technologists. These early experiences form a foundation for more advanced engagement with fairness, accountability, and bias in later courses, including algorithms, artificial intelligence, machine learning, bioinformatics, software engineering, and computing ethics seminars. Each

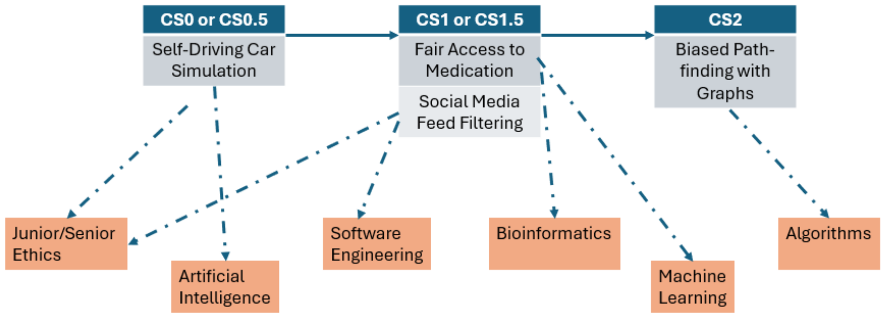


Figure 1: A curricular framework for integrating ethics labs in early CS courses. Labs in CS0–CS2 align with technical content and prepare students for ethical reasoning in advanced topics such as algorithms, artificial intelligence, machine learning, bioinformatics, software engineering, and computing ethics seminars.

lab is intentionally designed to be more than just a coding assignment; it is a site of ethical reasoning in action. Students are not simply writing code; they are evaluating the consequences of their decisions. Whether building a decision function for a self-driving car, filtering content in a social media feed, or analyzing fairness in healthcare data, students are encouraged to ask: Whose values are being encoded? What are the consequences of this design?

To foster these discussions, we recommend that ethics labs be conducted in pairs or small groups. This collaborative format encourages students to talk through their assumptions, confront competing interpretations of fairness or harm, and discover that ethical reflection is often messy, negotiated, and incomplete. These conversations allow students to see that code is not neutral and that how we write it can shape who is included, who is excluded, and who bears the burden of our technical systems.

The following ethics labs have been implemented in our early computer science curriculum. For each, we provide: (1) its curricular placement, (2) the ethical questions it raises, (3) the technical concepts it reinforces, and (4) how it lays groundwork for future coursework. While each lab stands on its own, all emphasize peer dialogue, structured reflection, and the development of ethical fluency through active programming.

Self-Driving Car Simulation

Ethical Focus: Fairness, bias, accountability in autonomous decision-making
Technical Concepts: Conditionals, functions, Boolean logic
Placement: Introduction to CS (CS0 or CS0.5)
Future Courses: AI, Junior/Senior Ethics Seminars

Students implement a decision function for a self-driving car navigating brake-failure scenarios, where they must weigh factors such as age, ability, legality, and number of lives. The simulation generates outcomes based on students' logic and displays comparative statistics (see Figure 2) against class or average values. This visual feedback surfaces differences in moral assumptions and encourages students to reflect on fairness, accountability, and unintended bias. Structured discussion and peer comparison deepen the exploration of ethical design choices in autonomous systems. Preliminary work on this lab was previously presented [10]. This lab builds awareness of ethical uncertainty early, laying a foundation for later conversations in AI and ethics-focused electives.

Fair Access to Medication

Ethical Focus: Algorithmic fairness, proxy attributes, equity in healthcare
Technical Concepts: Object-oriented programming, classification, confusion matrix metrics
Placement: Mid-level Programming (CS1 or CS1.5)
Future Courses: Machine Learning, Bioinformatics

Students use object-oriented programming to analyze algorithmic fairness in synthetic healthcare data. They apply four fairness definitions to compare treatment approval rates across demographic groups and reflect on how the same model can yield different outcomes depending on the metric. Bar charts (Figure 3) are generated to visualize differences in approval rates, true positive rates, and false positive rates across groups, helping students interpret the implications of each fairness definition. Lab discussions explore proxy attributes

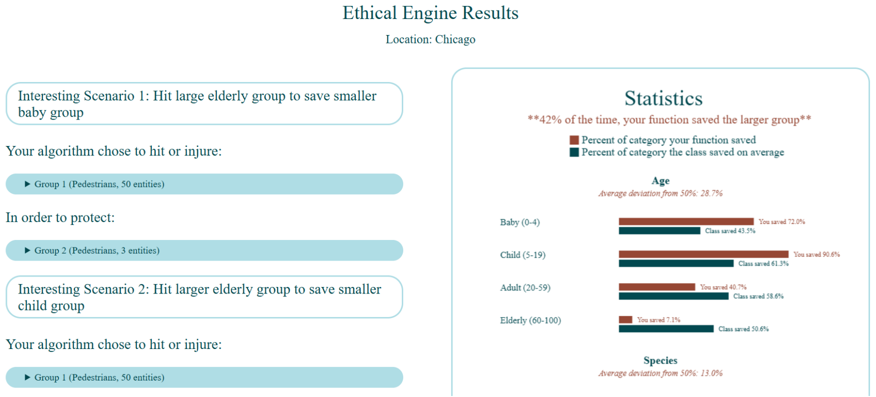


Figure 2: Ethical engine output comparing survival outcomes across attributes (e.g., age, gender expression, race) between the student's decision function and class averages. This visual encourages reflection on how different values may be encoded through code.

and trade-offs between fairness and accuracy. This lab anticipates fairness challenges students will revisit in Machine Learning and Bioinformatics.

Social Media Feed Filtering

Ethical Focus: Content moderation, freedom of expression, algorithmic bias
Technical Concepts: Lists, hash tables, basic filtering logic
Placement: Mid-level Programming (CS1 or CS1.5)
Future Courses: Junior/Senior Ethics, Software Engineering

In this lab, students simulate content filtering using flagged keywords and engagement-based algorithms. They observe how decisions affect visibility across users and groups. Through analysis and discussion, students explore how seemingly neutral filters may suppress marginalized voices or trigger unintended bias, echoing real-world debates on moderation, censorship, and algorithmic transparency. This lab surfaces ethical questions relevant to real-world systems and prepares students for later design discussions in software engineering and ethics-focused courses.

Biased Path-finding with Graphs

Ethical Focus: Structural bias, equitable access, urban inequality
Technical Concepts: Dijkstra's algorithm, edge weights, graph representations

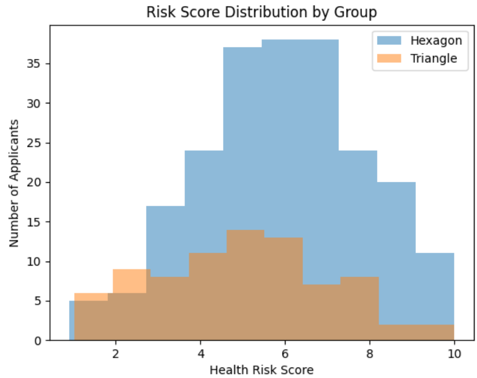


Figure 3: Risk score distribution by group. The histogram compares the health risk scores of applicants in the Hexagon and Triangle groups. Although both groups span similar score ranges, differences in distribution may contribute to disparities in approval decisions when a uniform threshold is applied. Students use this visualization to explore how group-level differences can impact fairness in algorithmic decision-making.

Placement: Data Structures (CS2)
Future Courses: Algorithms

Students use Dijkstra's algorithm to simulate urban navigation and experiment with modifying edge weights to reflect traffic, safety, or perception. They examine how design choices can reinforce structural inequality, using historical and contemporary examples like the Rondo neighborhood displacement[13]. Students discuss how routing algorithms encode values and impact communities. Later, in an Algorithms course, students can revisit path finding when comparing Dijkstra's with Bellman-Ford, allowing them to revisit and expand upon the design and ethical questions first raised in this lab.

5 Challenges and Status

A common concern about dedicated ethics labs is the already crowded nature of introductory CS courses. Instructors may worry that adding ethics content comes at the expense of core material. However, these labs are not replacements; they reinforce technical concepts like control flow, object-oriented design, and data structures. By embedding ethical reasoning into problem-solving, labs serve a dual purpose: teaching students to code and to reflect on what their code does in the world. Rather than being a distraction, ethics laboratories deepen the engagement with foundational CS skills.

Some students or faculty may initially view ethics as outside the “real work” of coding. Embedding ethical reasoning into implementation challenges that involve debugging, testing, and iteration helps reveal how technical and moral decisions are intertwined. We acknowledge that ethics discussions add some time to lab work, but the contextual framing mirrors what is typically required for real-world problems. In practice, the relevance of these scenarios often increases student investment, making technical material more accessible, not less.

Faculty may also feel unprepared to lead ethics discussions. Labs include instructor guides, sample responses, and structured prompts to support facilitation - no philosophy background required. Collaboration with humanities colleagues can further strengthen implementation.

Assessment is another concern. To support consistent grading,even by TAs, we provide structured rubrics aligned with lab goals. These include both factual criteria (e.g., accurate calculations) and ethical reasoning (e.g., justifying trade-offs). See Table 1 for an example. Rather than grading a “correct” moral stance, students are evaluated on how well they reason through competing values using evidence and course concepts. This approach supports scalability, even in large-enrollment courses.

Table 1: Sample Rubric for the Fair Access to Medication Lab

Criterion	**Meets Expectations**	**Minimally Meets Expectations**	**Does Not Meet Expectations**
Correct Implementation of Fairness Metrics	Student correctly implements all four fairness definitions and generates accurate approval rate, TPR, and FPR by group.	Student implements most metrics correctly but has minor errors in calculation.	Student implements few or no metrics correctly; major errors in logic.
Evaluation of Fairness Trade-offs	Student articulates a well-reasoned comparison of fairness definitions, noting potential benefits and harms for each, and connects choices to equity in healthcare.	Student compares fairness definitions but shows limited depth or clarity in evaluating trade-offs.	Student fails to compare fairness definitions or does so with no ethical justification.

Recognizing this gap in ethical engagement, we are beginning implementation of dedicated ethics labs in our introductory computer science sequence starting in Fall 2025. These labs will be integrated into first- and second-year technical courses, including introductory programming and data structures.

Each lab is designed to present students with a real-world scenario, guide them through coding and reflection, and prompt discussion among peers. Our goal is to build a scaffolded, iterative model that aligns with students' technical development, introducing ethical reasoning early and reinforcing it as students progress. This approach is intended to evolve across the curriculum, serving not only as a pedagogical tool for individual courses, but as a foundation for sustained ethical engagement throughout the major.

6 Conclusion and Call to Action

If we want students to graduate not only as skilled coders but also as responsible technologists, ethical reasoning must be part of their earliest computing experiences. The traditional approach, reserving ethics for a standalone course or brief module later in the curriculum, misses a critical opportunity to shape how students think about the consequences of their code from the very beginning.

This paper presents a curricular framework for integrating structured ethics labs across the early CS sequence. These labs are not isolated interventions but part of a broader scaffolding that aligns ethical reflection with technical progression, from foundational skills like control flow and data structures to later work in algorithms, AI, and software design. These labs combine hands-on programming with structured ethical reflection, peer discussion, and iterative decision-making. By embedding ethical questions directly into coding practice, they help students recognize that algorithms are not neutral and that the systems they build reflect values, priorities, and trade-offs.

We encourage departments to consider piloting a single ethics lab in an early course such as introductory programming or data structures. Instructors do not need to be experts in ethics to facilitate these experiences; well-designed lab guides, structured prompts, and support from humanities colleagues can make implementation accessible. Even one lab can serve as a meaningful entry point, prompting students to consider not just how to code, but how their code affects the world.

This framework is not the only path forward, but it is a practical, scalable, and pedagogically-aligned way to help students develop habits of ethical inquiry alongside technical fluency. By starting early and embedding reflection within practice, we can help ensure that students leave our classrooms ready not just to build technology, but to build it responsibly.

References

[1] Alexander W. Astin. "Student Involvement: A Developmental Theory for Higher Education". In: *Journal of College Student Personnel* 25.4 (1984), pp. 297–308.

[2] William T. Branch and Maura George. "Reflection-based Learning for Professional Ethical Formation". In: *AMA Journal of Ethics* 19.4 (2017), pp. 349–356.

[3] Noelle Brown, Sara Nurollahian, and Elaiane S. Wiese. "Evaluating Micro-Insertion as a Method for Teaching Responsible Computing". In: *ACM Transactions on Computing Education* 25.1 (2025), Article 9.

[4] Heidi B. Carlone and Angela Johnson. "Understanding the Science Experiences of Successful Women of Color: Science Identity as an Analytic Lens". In: *Journal of Research in Science Teaching* 44.8 (2007), pp. 1187–1218. DOI: 10.1002/tea.20237.

[5] Arthur W. Chickering and Linda Reisser. *Education and Identity.* 2nd ed. Jossey-Bass, 1993.

[6] Sarah Doore et al. "Experience Report: Using Narratives to Teach Responsible Computing in the U.S. and Nigeria". In: *Proceedings of the 56th ACM Technical Symposium on Computer Science Education (SIGCSE 2025).* ACM, 2025.

[7] Nancy J. Evans, Deanna S. Forney, and Florence Guido-DiBrito. *Student Development in College: Theory, Research, and Practice.* Jossey-Bass, 1998.

[8] Casey Fiesler et al. "Integrating Ethics into Introductory Programming Classes". In: (2021), pp. 987–993. DOI: 10.1145/3408877.3432510. URL: https://doi.org/10.1145/3408877.3432510.

[9] Barbara J. Grosz et al. "Embedded EthiCS: Integrating Ethics Across CS Education". In: *Communications of the ACM* 62.8 (2019), pp. 54–61. DOI: 10.1145/3341222.

[10] Tony Krystofiak et al. *Ethics in Motion: Interactive Software for Ethical Decision-Making in Autonomous Vehicles.* Poster presented at SIGCSE 2025. 2025.

[11] T. J. Landon and J. C. Schultz. "Exploring Rehabilitation Counseling Supervisors' Role in Promoting Counselor Development of Ethical Fluency". In: *Rehabilitation Counseling Bulletin* 61.4 (2018), pp. 237–246.

[12] Nevitt Sanford. *Where Colleges Fail: A Study of the Student as a Person.* Jossey-Bass, 1967.

[13] Alisha J. Volante. *The Rondo Neighborhood & African American History in St. Paul, MN: 1900s to Current: A Survey of Selected Sources*. Tech. rep. KNCBR #1402. Prepared in partnership with Rondo Avenue Inc. Accessed April 12, 2025. Center for Urban and Regional Affairs (CURA), University of Minnesota, 2015. URL: https://conservancy.umn.edu/server/api/core/bitstreams/de6a81f3-04bc-4275-b18c-b27a6902ca59/content.

Enhancing Computer Science Education Through Immersive Geometry-Based Approaches to Linear Algebra*

Sheldon Henriques[1], Shaun Lewis[1], Amos Chavan[1],

Abbas Attarwala, Ph.D.[1], Paul Viotti, Ph.D.[2] and Ed Lindoo, Ph.D.[3]

[1]Computer Science Department
California State University
Chico, CA, 95973
svhenriques@csuchico.edu, sdlewis@csuchico.edu, anchavan@csuchico.edu
aattarwala@csuchico.edu

[2]Political Science Department
California State University
Chico, CA, 95973
pviotti@csuchico.edu

[3]Computer Information Systems
Anderson School of Business
Regis University
Denver, CO, 80221
elindoo@regis.edu

Abstract

In this paper, we present a mixed reality (MR) application (app) developed to support linear algebra learning through interactive 3D visualization in a computer science course. Built for Meta Quest 3S, the app allows users to create and manipulate vectors, lines, and planes using hand tracking or controllers. A key feature is the geometric visualization of the Reduced Row Echelon Form (RREF) process, where users can see how elementary row operations transform planes without changing the solution to the underlying system of linear equations. This app transforms abstract matrix operations into intuitive geometric visualizations, making complex linear algebra concepts more accessible to computer science students. Created through an independent study in computer science, the tool is designed to complement topics typically covered in applied linear algebra courses. Initial classroom implementation suggests that it is particularly effective for visual and kinesthetic learners, demonstrating the potential of MR to enhance both comprehension and engagement in undergraduate computer science education.

1 Introduction

Linear algebra is a foundational subject in computer science, engineering, and mathematics, yet it remains one of the most conceptually challenging courses for many undergraduate students [5, 12, 6]. One key reason is the abstract nature of its core operations, which are often taught symbolically through matrix manipulation without sufficient emphasis on their geometric interpretations [20]. As a result, students may learn to perform procedures like RREF but often struggle to grasp why these operations work and how they affect geometrically the solution set of the underlying system of equations. Research and teaching experience suggest that students grasp the core ideas of linear algebra most effectively when these concepts are grounded in the familiar geometry of two and three dimensions (3D). When students build confidence through visual and spatial understanding in lower dimensions, they are better prepared to extend those ideas to higher-dimensional and more abstract settings [2].

This work addresses that gap by introducing an MR application that enables students to interactively visualize systems of linear equations in 3D. Within the immersive environment, users can create and manipulate planes, lines, and vectors using their hands or controllers, and observe the evolving geometric interpretation of a system as it is reduced to RREF via elementary row operations. Students observe that although the individual planes are altered, the solution set—whether a point, line, or plane—remains unchanged. This reinforces the conceptual purpose of elementary row operations: they simplify the system while preserving its solutions.

The introduction of this MR-based tool has proven especially valuable for students who benefit from visual-spatial learning, as well as those who struggle with abstract reasoning or lack experience with symbolic mathematical notation. By translating linear algebra into an interactive geometric format,our app enhances conceptual accessibility and creates a deeper comprehension of the material. When used in conjunction with a step-by-step numerical breakdown of the RREF algorithm in computational environments such as MATLAB, students are able to bridge geometric intuition with algorithmic execution. This dual-modality approach not only reinforces understanding in three-dimensional contexts but also equips learners with the abstract reasoning skills necessary to extend these concepts to higher-dimensional spaces where direct visualization is no longer practical. In the rest of our paper, we first present a review of the relevant literature in Section 2. We then present the design and implementation of the MR application and demonstrate its pedagogical rationale in Section 3, and reflect on its early use with a group of former and current students in Section 4. Using a computer science course of applied linear algebra, we show how immersive embodied experiences can make abstract mathematical concepts more intuitive, inclusive, and meaningful.

2 Literature Review

Linear algebra forms a foundational component of computer science education, serving as the mathematical backbone for graphics processing, machine learning algorithms, and computational modeling. Students often struggle with concepts such as vector spaces, linear transformations, and eigenvalues. They may successfully perform computational procedures, yet fail to grasp their geometric meaning or conceptual significance. Those working on mathematical pedagogy have developed embodied learning approaches that address these challenges through visual-geometric representations, immersive technologies, inquiry-based instruction, and supporting cognitive theories. Visualization and technology can help make abstract ideas in linear algebra more concrete by letting students see and interact with mathematical structures in three dimensions [16].

2.1 Visual and Geometric Perspectives in Linear Algebra Instruction

A recurring theme in contemporary scholarship on linear algebra pedagogy is the growing emphasis on integrating geometric intuition alongside the traditional algebraic formalism. This pedagogical shift reflects a broader recognition that abstract symbolic methods, while essential, may not fully support conceptual understanding for all learners—particularly in the context of complex transformations and multidimensional reasoning. [20] highlights that traditional courses often over-emphasize symbolic manipulation while neglecting geometric interpretation, leaving students with inadequate conceptual understanding. The paper argues that integrating a geometric perspective, by consistently linking algebraic concepts to their spatial interpretations in $\mathbb{R}^2$ and

$\mathbb{R}^3$, can enhance students' interest and comprehension. Through illustrative examples involving linear dependence, solution structures of linear systems, and orthogonalization processes, [20] demonstrates how the pedagogical approach of connecting numbers and shapes helps students transform abstract mathematical concepts into tangible spatial geometric understanding.

These findings align with recommendations from the Linear Algebra Curriculum Study Group (LACSG 2.0), which advocates leveraging technology in teaching and emphasizes concept-driven instruction over purely algorithmic approaches. The report notes that dynamic geometry software can support conceptual understanding at the introductory level—such as visualizing how a linear transformation scales or rotates vectors—while also highlighting that the effectiveness of geometric approaches depends on the type of course and student population [16].

2.2 Immersive Learning Environments: Virtual and Augmented Reality

Building on the recognized value of visualization in mathematics education, recent studies have increasingly investigated the use of immersive technologies—particularly virtual and augmented reality—as tools to improve conceptual understanding in linear algebra. These technologies offer interactive and embodied experiences for students that render abstract mathematical constructs such as vectors, subspaces, and linear transformations in $\mathbb{R}^n$ as perceptible objects within a navigable three-dimensional space. By situating algebraic concepts in a spatial context, Virtual Reality (VR) and Augmented Reality (AR) environments serve to bridge the gap between formal symbolic manipulation and intuitive geometric reasoning, thereby supporting deeper learning and engagement

[13] introduced an immersive VR module in a first-year linear algebra course, focusing on affine geometry concepts within a virtual environment. Students used VR headsets to manipulate points, vectors, and planes in 3D space, observing in real time how changes in coordinates affect geometric configurations. The intervention was designed to address spatial visualization deficiencies, and student surveys indicated high engagement, with more than 90% requesting additional VR-based geometry activities.

[4] developed MathVR, an application that allows learners to enter a 3D coordinate system and perform vector arithmetic operations (addition, subtraction, projection and cross product) using their hands. Users can physically manipulate vectors and immediately see operation results updated in real-time, though formal evaluation of the system's educational efficacy is still planned.

Empirical evidence validates the effectiveness of these approaches. For example, [10] conducted a comparative study of a metaverse-based linear algebra platform and conventional teaching. Students who learned with the immersive platform outperformed those in traditional settings on knowledge tests about 3D linear transformations and reported greater enjoyment during the learning process.

For algebraic problem-solving, [19] developed an AR escape room game designed to teach systems of linear equations. Students used mobile devices to view AR content overlayed on a physical Merge Cube, solving interactive puzzles about linear equations. The AR group demonstrated significantly improved performance in solving linear systems, and qualitative data indicated that the students found the approach more interesting and engaging than traditional instruction.

These studies underscore a clear trend: immersive media turn linear algebra from a passive, symbolic endeavor into an active, spatially grounded experience. This approach resonates with LACSG 2.0's recommendation to "take advantage of technology in teaching, [by] motivating concepts with applications" [16], extending this vision into immersive VR and AR environments that make abstract concepts more spatially accessible.

2.3 Conceptual and Inquiry-Based Teaching Strategies

In parallel with visual and immersive techniques, there is continued momentum toward inquiry-oriented and conceptually-focused pedagogies in linear algebra. These approaches focus on actively engaging students in making meaning of definitions and theorems rather than having them memorize procedures. For example, a multi-institution study by [7] provided compelling evidence for the efficacy of inquiry-based instruction. In their experiment with 19 introductory linear algebra classes, sections taught using Inquiry-Oriented Linear Algebra (IOLA) materials emphasized student exploration through guided problems and discussion. The Post-assessment results showed that the inquiry-based sections outperformed the traditional sections on both conceptual understanding and procedural skills. [7] reported the most significant learning gains in classrooms led by instructors with prior experience using IOLA materials, suggesting that instructor familiarity with inquiry-oriented approaches improves student outcomes. These findings reinforce a growing consensus that active learning approaches are critical for developing robust conceptual knowledge. The LACSG 2.0 report advocates for a move away from rote teaching and encouraging conceptual reasoning in linear algebra classrooms [16].

Even technology rich interventions are most effective when embedded in an active learning context, for example, students exploring VR modules in groups rather than passively watching animations. The trend is toward blending innovative tools with student-centered pedagogy, using visualization as a catalyst for inquiry and conceptual dialog.

2.4 Theoretical Foundations and Context for Immersive Mathematics Learning

Recent scholarship has increasingly emphasized the pedagogical value of immersive and visual technologies in linear algebra instruction, particularly when grounded in contemporary models of student engagement and peer learning. Although foundational theories such as the work of [3]' on self-efficacy and [18] communities of practice continue to provide important context—highlighting the role of mastery experiences and socially mediated meaning-making—recent studies have extended these ideas to the realm of immersive learning environments. For example,[10] and [13] demonstrate that immersive 3D environments can lead to measurable improvements in understanding linear algebra concepts, with [13] showing gains in comprehension of eigenvalue and eigenvectors through individual VR experiences. These findings suggest that the immediacy and interactivity of 3D environments can create deeper conceptual engagement. Such results resonate with [7] work, which underscores the importance of pedagogical experience in effectively leveraging student-centered tools. In this light, earlier theories serve not as primary explanatory models, but as conceptual scaffolding that helps to interpret the mechanisms behind the success of emerging VR-based learning platforms in mathematics education. Throughout the literature reviewed, a clear message emerges: to improve learning in linear algebra, educators must make the invisible visible, whether through geometric diagrams, immersive 3D worlds, or carefully crafted explorations. Visual approaches provide intuition, immersive technologies provide interaction, and inquiry-based strategies ensure that students actively construct understanding rather than passively receiving it.

Our proposed MR application for teaching linear algebra in computer science is designed at the nexus of these theoretical frameworks and pedagogical trends. By immersing students in MR environments where they can visualize and manipulate vectors and perform linear transformations, such approaches capitalize on the documented benefits of VR/AR for engagement and spatial understanding [10, 4, 13]. These immersive environments can also encourage exploration, aligning with inquiry-oriented approaches proven to yield conceptual gains [7].

Our app serves as a platform for students to embody linear algebra concepts, combining visual intuition with conceptual inquiry and immersive interaction. This integrated approach is well-supported by learning theory and positions our work as a natural next step in the evolution of linear algebra pedagogy, contributing to evidence that making linear algebra more visual, interactive, and student-centered can dramatically improve conceptual understanding.

3 Interactive Learning in MR

We describe our app as an MR experience rather than VR because it integrates digital mathematical objects into the user's physical environment. The app begins by allowing users to define a boundary in their real-world space, which establishes a fixed origin for the 3D coordinate system. The planes, vectors, and lines are then visualized in relation to this physical anchor, allowing students to move around and view the objects from different angles while still seeing their real surroundings. This merging of virtual elements with the real world supports spatial reasoning without isolating users from their environment. A key feature of our MR app is the step-by-step visualization of the RREF process. Rather than relying solely on symbolic manipulation, students see how row swaps, scaling, and row replacements transform the geometry of the system. These operations reorient the planes without changing the solution set, helping students understand both the mechanics and the meaning of RREF. To the best of our knowledge, this is the first time such a geometric unfolding of the RREF process has been implemented in MR using Meta Quest 3S hardware. To accomplish this, we developed the app with computer science students in our Applied Linear Algebra course for computer science using the C# programming language and the Unity app SDK.

In Figure 1, we show an example of a system of three linear equations representing three planes that intersect at a unique point. The left panel depicts the initial system, while the right panel shows the final step of the RREF. In the MR environment, students can interactively step through the intermediate stages using a "Next RREF" and "Prev RREF" button, observing how the planes shift with each operation. In Figure 1, we also show the augmented matrix corresponding to the initial system of linear equations. The matrix is $\begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & -4 & -2 & 1 \\ -2 & 4 & -5 & 1 \end{bmatrix}$, which represents the planes $x+y+z=1$, $2x-4y-2z=1$, and $-2x+4y-5z=1$. These three planes intersect at a unique point, shown in the figure as $(0.93, 0.36, -0.29)$. On the right side of the same figure, we show the geometry after applying the RREF algorithm. Although the orientation of the planes has changed, the solution set remains the same. The corresponding RREF matrix is $\begin{bmatrix} 1 & 0 & 0 & 0.93 \\ 0 & 1 & 0 & 0.36 \\ 0 & 0 & 1 & -0.29 \end{bmatrix}$, which corresponds to the equations $x = 0.93$, $y = 0.36$, $z = -0.29$. This reinforces the idea that elementary row operations reorient the system geometrically without altering its solution.

Figure 1: Initial (before RREF) and final (after RREF) states of a system of three planes intersecting at a unique point.

In Figure 2, we present a system with infinitely many solutions, where two planes intersect along a line. Here we show the parametric equation of the line where the two planes intersect. As before, the initial and final configurations are shown, with full stepwise visualization available in the app. [8] investigated how students in lower secondary school reason with 3D geometry problems when those problems are presented using two-dimensional representations, such as cube drawings on paper or classroom boards. Their findings show that many students rely heavily on visual cues from these 2D projections, often leading to incorrect conclusions. For example, a large proportion of students misjudged angles or constructed flawed geometric arguments based on flattened or distorted visualizations of the cube. The authors highlight that some students required alternative perspectives or representations to understand spatial relationships correctly, while only a few were able to reason abstractly about 3D structures beyond the visual information provided.

We hypothesize that a similar phenomenon may occur in the context of linear algebra, where students overgeneralize the familiar two-dimensional line equation $y = mx + c$ to higher dimensions. This tendency to overextend reasoning from 2D cases may partly explain why students struggle to grasp the more abstract vector and parametric forms of lines in 3D. These observations motivated the development of our MR app, which aims to help students visualize planes, vectors, and RREF forms in 3D, and thus overcome the limitations of static 2D representations in traditional instruction.

Although this visualization helps clarify that in 3D, a line is more accurately described as the intersection of two planes and is typically expressed in parametric form. Seeing this relationship visually reinforces the distinction and deepens the students' understanding of how lines behave in a 3D space.

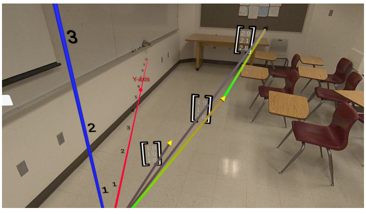
Figure 2: Initial (before RREF) and final (after RREF) states of a system of two planes intersecting on a line.

Figure 3 illustrates a system with no solution. Here, the final RREF step reveals a contradiction, which is reflected in a message shown within the MR environment that indicates that no solution exists. In this case, the geometry ceases to be meaningful in relation to a solution set, reinforcing the algebraic outcome through a visual cue.

Figure 3: Initial and final states of a linear system with three parallel planes and no solution. After RREF, the app displays "No solution," and the geometry becomes irrelevant. The initial view clearly shows that the planes do not intersect.

Beyond solving systems of equations, the app also provides interactive visualizations of fundamental vector operations. In Figure 4, we show a participant performing vector addition by selecting two vectors and dynamically observing the resulting vector. Research in mathematics education highlights that geometric and visual representations significantly improve students' understanding of vector operations such as addition and span of a vector. Teaching linear algebra without concretizing concepts through visualization can drive students to memorize definitions and techniques, leaving their understanding at a purely computational level [1]. Interactive visualizations help address common misconceptions. For example, [1] found that students mistakenly believed that a single nonzero vector could span $\mathbb{R}^2$ until they used dynamic software to explore scalar multiples, revealing the true one-dimensional nature of the span. [9] demonstrated that an interactive system using embodied touch-based vector manipulations led to improved cognitive engagement and improved integration of geometric and algebraic reasoning approaches in vector concepts. The experimental group showed significantly higher completion rates on vector tasks (2.61 times more likely to complete all questions) and demonstrated more balanced use of both geometric and algebraic reasoning strategies. These tools promote active learning and help students connect algebraic procedures to geometric intuition. [17] found that many undergraduates could perform vector operations but struggled with visual interpretation. [14] found that visualization interventions significantly improved mathematics learning outcomes, including both conceptual understanding and long-term retention, compared to instruction without visualizations.

Our app builds on this research by offering interactive geometric experiences of vector addition and span. As shown in Figures 4 and 5, students can dynamically manipulate vectors, observe real-time results, and visually explore linear combinations. This aligns with the findings that interactive visual feedback supports the conceptual development of geometric relationships, improves student engagement and motivation, and can make complex mathematical concepts accessible to diverse learners [11].



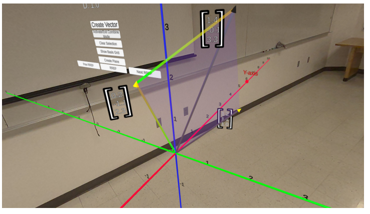
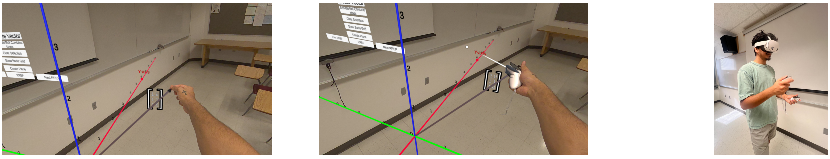


Figure 4: Visualization of vector addition in our app. The first figure demonstrates addition of vectors [1.9, 1.1, 1.8] and [1.0, 1.0, 1.0], yielding the resultant vector [2.9, 2.1, 2.8]. The second figure highlights the geometric interpretation of vector addition as a parallelogram formed by two vectors.



(a) Span of a vector (b) Span of a vector (c) With Meta Quest 3S

Figure 5: Visualizing span of a vector and participant interaction.

Combining visual, physical, and symbolic learning in a classroom helps reach a wider range of students, especially those who learn better with concrete examples and spatial thinking. As highlighted in a survey of linear algebra education research [15], geometric representations, particularly when coordinated with algebraic and symbolic reasoning, can support student understanding of concepts such as span and linear independence.

4 Initial User Feedback and Planned Improvements

To gather initial feedback on usability and educational value, we conducted a pilot study with ten students who had completed an applied linear algebra course. Participants explored concepts such as vector addition, span, and the RREF process using the app, then submitted responses via a structured Google Form. Likert-scale results (1 = Strongly Disagree, 5 = Strongly Agree) are summarized in Table 1. Although the sample size is small, this pilot study reflects the current development stage of the application. The primary goal was to collect formative feedback to inform design and pedagogical improvements. As the app matures and becomes more fully integrated into classroom instruction, we plan to conduct broader studies with larger and more diverse student populations to evaluate its impact on learning outcomes at scale.

Table 1: Number of participants (out of 10) selecting each Likert scale option for each question. A value of 0 indicates no participants selected that option. Rows sum to 10 responses per question.

Question	Ratings					Mean	Std. Dev
	1	2	3	4	5		
The MR tool helped me better understand the concepts of vectors and planes in 3D.	0	0	0	4	6	4.6	0.49
The reduced row echelon form visualizations were clear and useful.	0	0	0	3	7	4.7	0.46
The interaction and navigation within the MR environment were intuitive.	0	1	0	3	6	4.4	0.84
I would recommend this MR experience to other students learning linear algebra.	0	0	0	1	9	4.9	0.30
Overall, I found the MR session to be engaging and educational.	0	0	1	1	8	4.7	0.67

Qualitative feedback emphasized the value of visualization. One student shared that the MR experience *made tough concepts easier to understand by showing them visually,* especially noting that *the RREF part was really clear* with step-by-step changes in the augmented matrix. Another commented that *seeing the parametric line appear where the planes intersect helped me finally get what the solution really means,* adding that it helped them *see how everything fits together in 3D.* Feedback from the pilot evaluation highlighted the need for better color schemes, clearer vector labeling, and a guided tutorial. The participants also recommended adding sliders for precise vector control, noting that the current sliders are too sensitive and move too quickly. Suggestions also included the use of distinct plane configurations for clarity. Future updates will address these points by including labeled objects, improved visuals, less sensitive sliders, and an optional tutorial or voice-over. The evaluation revealed both strengths and usability gaps, setting the stage for a broader study during the next applied linear algebra course.

5 Conclusion

This paper introduced a MR application that enhances linear algebra education through interactive 3D visualizations of vectors, planes, and systems of equations. The app, written by computer science students using C# created a notable feature; the geometric visualization of the RREF algorithm, helping users connect symbolic steps to spatial transformations. Initial feedback indicates that the app makes abstract concepts more intuitive and engaging, particularly for visual learners. Future plans include classroom deployment, expansion of features to cover advanced topics, and a broader evaluation of its educational impact. The work highlights MR's potential to make linear algebra more accessible and meaningful.

Acknowledgment

We gratefully acknowledge the use of OpenAI's ChatGPT for proofreading, grammatical checks, and other text editing tasks.

References

[1] Cahit Aytekin and Yasemin Kiymaz. "Teaching Linear Algebra Supported by GeoGebra Visualization Environment." In: *Acta Didactica Napocensia* 12.2 (2019), pp. 75–96.

[2] Thomas Banchoff and John Wermer. *Linear algebra through geometry.* Springer Science & Business Media, 1993.

[3] Albert Bandura. *Self-efficacy: The exercise of control.* New York: W.H. Freeman and Company, 1997.

[4] Matt A. V. Cabanag. “MathVR: Teaching Vector Arithmetic Using Virtual Reality”. In: *SIGGRAPH Asia 2023 Educator’s Forum.* Association for Computing Machinery, 2023, pp. 1–3. DOI: 10.1145/3610540.3627012.

[5] Jean-Luc Dorier and Anna Sierpinska. “Research into the teaching and learning of linear algebra”. In: *The teaching and learning of mathematics at university level: An ICMI study.* Springer, 2001, pp. 255–273.

[6] Ghislaine Gueudet-Chartier. “Using geometry to teach and learn linear algebra”. In: *Research in collegiate mathematics education* 6 (2006), pp. 171–195.

[7] Muhammad Qadeer Haider and Christine Andrews-Larson. “Examining learning outcomes of inquiry-oriented instruction in introductory linear algebra classes”. In: *International Journal of Education in Mathematics, Science, and Technology* 10.2 (2022), pp. 341–359. DOI: 10.46328/ijemst.2096.

[8] Keith Jones, Taro Fujita, and Susumu Kunimune. “Representations and reasoning in 3-D geometry in lower secondary school”. In: *Proceedings of the 36th Conference of the International Group for the Psychology of Mathematics Education (PME36).* Ed. by Tien-Yeh Tso. Vol. 2. Taipei, Taiwan: PME, 2012, pp. 339–346.

[9] DurgaPrasad Karnam et al. “Touchy Feely Affordances of Digital Technology for Embodied Interactions Can Enhance ’Epistemic Access’”. In: *2019 IEEE Tenth International Conference on Technology for Education (T4E).* IEEE. 2019, pp. 114–121.

[10] Yicong Li et al. “Enhancing Linear Algebra Education with AI-Generated Content in the CityU Metaverse: A Comparative Study”. In: *Proceedings of the International Symposium on Emerging Technologies for Education (SETE 2024).* Lecture Notes in Computer Science, vol. 15589. Springer, 2025, pp. 3–16. DOI: 10.1007/978-981-96-4407-0_1.

[11] Hai Liang and Kamran Sedig. “Can interactive visualization tools engage and support pre-university students in exploring non-trivial mathematical concepts?” In: *Computers & Education* 54.4 (2010), pp. 972–991.

[12] Ragnhild Johanne Rensaa, Ninni Marie Hogstad, and John Monaghan. “Perspectives and reflections on teaching linear algebra”. In: *Teaching Mathematics and its Applications: An International Journal of the IMA* 39.4 (2020), pp. 296–309.

[13] José Luis Rodríguez. “Using virtual reality to teach linear algebra with a focus on affine geometry”. In: *Proceedings of the Asian Technology Conference in Mathematics (ATCM 2022)*. Mathematics and Technology LLC, 2022, pp. 218–230.

[14] Johanna Schoenherr, Anselm R Strohmaier, and Stanislaw Schukajlow. “Learning with visualizations helps: A meta-analysis of visualization interventions in mathematics education”. In: *Educational Research Review* (2024), p. 100639.

[15] Sepideh Stewart, Christine Andrews-Larson, and Michelle Zandieh. “Linear algebra teaching and learning: themes from recent research and evolving research priorities”. In: *ZDM* 51 (2019), pp. 1017–1030.

[16] Sepideh Stewart et al. “The Linear Algebra Curriculum Study Group (LACSG 2.0) Recommendations”. In: *Notices of the American Mathematical Society* 69.5 (2022), pp. 813–820. DOI: 10.1090/noti2479.

[17] Muhammad Reyza Arief Taqwa and Handy Faishal Rahim. “Students’ conceptual understanding on vector topic in visual and mathematical representation: a comparative study”. In: *Journal of Physics: Conference Series*. Vol. 2309. 1. IOP Publishing. 2022, p. 012060.

[18] Etienne Wenger. *Communities of Practice: Learning, Meaning, and Identity*. Cambridge: Cambridge University Press, 1998.

[19] Mireya Zapata et al. “Enhancing mathematics learning with 3D augmented reality escape room”. In: *International Journal of Educational Research Open* 7 (2024), p. 100389. DOI: 10.1016/j.ijedro.2024.100389.

[20] Min Zhang. “Illustrated Difficulties in Teaching Linear Algebra”. In: *Curriculum and Teaching Methodology* 7.1 (2024), pp. 184–189. DOI: 10.23977/curtm.2024.070127.

AI-Assisted Contextual Code Narration for the Visually Impaired*

Prajwal Narayanaswamy[1], Abbas Attarwala[1], Paul Viotti[2], Jaime Raigoza[1] and Ed Lindoo[3]
[1]Computer Science Department
California State University, Chico
Chico, CA, 95973
plnu@csuchico.edu, aattarwala@csuchico.edu, jraigoza@csuchico.edu
[2]Political Science Department
California State University, Chico
Chico, CA, 95973
pviotti@csuchico.edu
[3]Computer Information Systems
Anderson School of Business
Regis University
Denver, CO, 80221
elindoo@regis.edu

Abstract

This paper introduces a new methodology designed to enhance the accessibility of source code for computer programmers who are blind or visually impaired (BVI). Conventional screen readers, such as JAWS, typically interpret and vocalize code in a linear, word-by-word fashion, often failing to convey the contextual relationships and structural hierarchy fundamental to program comprehension. To overcome these limitations, we propose a system that leverages the OpenAI large language model (LLM), guided by carefully engineered prompts, to transform source code into structured, context-aware natural language descriptions. For example, rather than sequentially narrating deeply nested conditional statements line by line, our system first summarizes the structural context beforehand (e.g., identifying a nested conditional statement five levels deep) and subsequently details each condition clearly within this context. Our system, implemented as a Visual Studio Code extension, uses Eleven Labs Text-to-Speech (TTS) to read code aloud with two distinct voices. We designate a female voice, `Rachel`, to read each line of source code verbatim and a male voice, `Andy`, to provide contextual explanations. Although this approach is designed to significantly enhance BVI' comprehension of code structure and logic, providing a more intuitive and efficient auditory coding experience, it has not yet been formally evaluated with BVI individuals.

1 Introduction

Programming and software development are increasingly essential skills in numerous fields. However, BVI developers often encounter significant barriers due to the inherent limitations of traditional screen reader software, which typically vocalizes code in a linear, line-by-line fashion [9, 2]. This approach, although functional for basic textual navigation, struggles to convey the structural and logical context critical for comprehending complex programming constructs, such as deeply nested conditional statements and iterative loops.

To address these challenges, we propose a solution that takes advantage of recent advances in artificial intelligence (AI), with a particular emphasis on LLMs such as those developed by OpenAI, to improve the auditory rendering of source code. By employing meticulously designed prompt engineering techniques, our approach enables LLMs to generate structured, context-aware summaries of code that substantially enhance its comprehensibility when delivered through TTS systems. For example, when processing complex constructs, such as nested loops or conditional statements, the system initially conveys a high-level overview of the control structure, followed by a detailed narration of individual components. This method preserves contextual coherence and significantly improves the user's ability to cognitively parse and navigate programming logic in an auditory format. To accommodate diverse user preferences and usage scenarios, the proposed solution also offers an alternative mode in which BVI users can access the source code in a verbatim, line-by-line format without contextual augmentation. This functionality ensures greater flexibility, allowing BVI users to select between enriched contextual summaries and a direct, unmodified auditory representation of the code based on their individual needs or task requirements. Implemented as an extension within Visual Studio Code (VSCode), a widely adopted Integrated Development Environment (IDE), this tool integrates seamlessly into the existing workflows of developers, promoting both accessibility and ease of adoption. By bridging the gap between traditional screen readers and meaningful code comprehension, our approach empowers BVI programmers to engage more fully and efficiently in software development activities.

2 Literature Review

BVI individuals remain severely underrepresented in software development. In a large developer survey, only 2% of respondents reported having a disability of some kind, including visual impairments [9]. Researchers have identified significant accessibility barriers in current programming tools as a contributing factor to this underrepresentation [9].

Contemporary code editors and IDEs rely heavily on visual cues such as syntax highlighting, indentation, and graphical debuggers, which screen readers struggle to capture or interpret effectively. Potluri et al. [11] categorize the accessibility challenges of IDEs into four areas: *discoverability*, *glanceability*, *navigability*, and *alertability*. Glanceability refers to the ability to quickly skim and grasp code structure at a glance—something sighted developers achieve through visual scanning, while screen reader users must process information linearly. For example, sighted developers can quickly navigate through code using scroll, point, and click, while screen reader users have limited navigation options [11]. Subsequent research has confirmed that these challenges persist in modern development environments [6].

A recent systematic review of the literature by [9] confirms these persistent barriers, from code navigation and understanding of the state of the program to the lack of accessible debugging tools, that continue to hinder BVI students and developers. Without additional support, BVI programmers must expend excessive time and cognitive effort to comprehend others' code or keep track of program context, which in turn deters many from pursuing coding as a career or academic interest.

Efforts to address these challenges have focused on adapting or augmenting the development environment for non-visual interaction. [11] introduced *CodeTalk*, a Visual Studio plugin that provides auditory cues and structured code navigation for BVI developers. CodeTalk addresses glanceability challenges, specifically the difficulty of quickly obtaining overview information when navigating code linearly with a screen reader, by providing keyboard shortcuts that generate accessible tree views of code structure (e.g., namespaces, classes, and functions) and accessible lists of specific code elements. The plugin also announces IDE events (such as real-time compiler errors or debugging information) through audio cues that would otherwise go unnoticed by a screen reader user.

Another approach has been to redesign the editing paradigm itself: Grid Editor [6] is a recent system that represents source code in a two-dimensional spreadsheet-like grid, with each row and column denoting code lines and nesting levels, respectively. This structured layout, coupled with custom keyboard navigation and audio cues, enables BVI programmers to navigate and edit code more efficiently and accurately than in a plain text editor.

Likewise, researchers have explored making block-based visual programming languages usable through non-visual means. [16] replaced the complex spatial navigation required by block-based environments such as Blockly with a simple line-by-line navigation interface and reported that this significantly reduced task completion times and improved usability ratings in an evaluation with blindfolded blind participants. Despite these advances, many solutions require specialized hardware or custom environments (e.g., tactile displays, bespoke editors), and thus have seen limited adoption in mainstream programming practice [9]. The need remains for integrated accessibility features that can be easily deployed in popular coding tools to bridge the gap between sighted and visually impaired programmers.

The emergence of AI and LLMs offers new possibilities to improve programming accessibility. LLM-based assistants can generate natural-language descriptions of code behavior, summarize complex code sections, and answer clarifying questions about code, effectively acting as on-demand narrators. Recent research has begun to apply these capabilities to assist BVI developers. For example, [12] present *BLVRUN*, a command-line tool that intercepts runtime error traces and uses an LLM to generate concise, informative summaries of Python traceback errors. This helps BVI programmers debug faster by cutting through verbose error logs and highlighting the relevant causes of an error in plain language. In the context of education, [16] integrated a conversational AI (based on a code-capable LLM) into their accessible block-language environment to produce high-level summaries of the program's purpose. Their user study with blindfolded sighted participants found that those who received these AI-generated code summaries reported high levels of comprehension and reduced anxiety in subjective evaluations, although testing with actual blind users remains future work.

These AI-driven approaches represent a significant advancement over earlier accessibility solutions, such as CodeTalk's TalkPoints, which relied on predefined speech and non-speech audio cues for debugging [11]. Although CodeTalk's audio cues required manual configuration, contemporary LLM systems can dynamically generate contextual explanations without explicit setup. More broadly, LLM assistants such as GitHub Copilot and ChatGPT have been touted as equalizers in programming education. By offering immediate access to explanations, contextual hints, and documentation retrieval, these tools offer particular promise to learners who encounter barriers to traditional programming resources, including individuals with disabilities [7]. In this context, [7] observed that BVI developers are optimistic about the potential of AI coding assistants to improve their workflow by automating repetitive tasks and providing timely guidance, thus improving their productivity and coding efficiency. However, these benefits are highly dependent on accessible design. If implemented incorrectly, AI-based tools might introduce new frustrations for BVI users. For example, the study highlights that GitHub Copilot occasionally inundates users with excessive suggestions and creates difficulties in managing focus between the code editor and the assistant's output. Participants in that study emphasized the need for "AI timeouts" or controls to prevent AI from flooding the screen reader with information.

A recent study by [1] examined how BVI people use general-purpose AI chatbots such as ChatGPT in everyday tasks. Through qualitative interviews, the authors found that while BVI users frequently rely on these tools to gain information or solve problems (including coding questions), they often have to devise workarounds to navigate chatbot interfaces with a screen reader and must carefully verify AI outputs for accuracy. These findings highlight that the mere inclusion of AI assistance does not inherently guaranty accessibility. Instead, any AI-driven accessibility tool should strive to deliver support in a controlled, contextual manner that aligns with the user's normal coding habits, rather than disrupting them.

Our approach draws upon established insights from accessibility research while introducing a fundamental shift in how code narration is generated and delivered. CodeTalk [11] showed that BVI developers benefit from proactive information extraction, as screen reader users "must actively seek out information from various components of the IDE." CodeTalk provided accessible tree views and audio debugging cues (TalkPoints), but these required manual configuration by developers. Grid Editor [6] demonstrated the value of alternative code representations through coordinated text and grid views, though this required a dedicated environment separate from standard editors. Similarly, Stefik et al. [15] validated that auditory cues representing lexical scoping can improve program comprehension, using symbolic sounds to convey code structure.

Building on these foundations, our VS Code extension introduces two key innovations. First, rather than requiring manual configuration like CodeTalk's TalkPoints or using symbolic sounds like Stefik's scoping cues, we leverage OpenAI's LLM to automatically generate natural-language explanations of code structure and purpose in real-time. Second, we employ a dual-voice design for perceptual separation: a female voice reads the code verbatim, while a male voice provides a contextual explanation. This separation reduces cognitive load when processing complex code, addressing the glanceability challenges identified in prior work. Importantly, our extension operates seamlessly within the standard VS Code interface and activates on-demand during navigation, avoiding both the workflow disruption of dedicated environments and the "constant deluge of suggestions" observed by [7].

The need for such capabilities is underscored by CodeTalk's user study, where one participant noted: "I never knew how much information I was not getting because I was using a screen reader" [11]. This remark highlights how accessibility barriers can become normalized, emphasizing the importance of systems that actively reveal the structural and semantic aspects of code that sighted developers can access visually.

3 Implementation Overview

Our accessibility solution is implemented as a VS Code extension. This choice is driven by the editor's widespread adoption among developers, its ease of integration, and its flexible architecture. According to the Stack Overflow 2024 Developer Survey [14], VS Code is used by 74% of developers, making it the most popular IDE by a significant margin—more than double the usage of its nearest alternative, Visual Studio (29%). While other indices such as the PYPL Top IDE Index [4] rank Visual Studio higher in terms of tutorial search popularity (28.48% versus VS Code's 15.27%), they also highlight the rapid growth trend of VS Code over the past five years. This widespread usage ensures that an extension developed for VS Code has the potential to reach a large and diverse audience of developers.

Beyond popularity, VS Code offers architectural advantages that make it suitable for this project [17]. It provides a well-documented extension API that enables integration of custom features. This extensibility allows the extension to automatically process code upon file opening and trigger audio playback when users interact with specific lines. VS Code also supports a wide range of programming languages, including C++, which remains relevant in many domains. Targeting VS Code allows us to embed the extension in a familiar and widely used environment for C++ developers. Its cross-platform availability on Windows, macOS, and Linux further enhances accessibility, while its large and active community contributes valuable support and resources [18].

The core functionality of the extension involves processing C++ source code and generating audio representations. When a user opens a C++ file, the extension parses it line-by-line. For each line, two audio outputs are generated: a verbatim narration of the code spoken in a female voice (which we refer to as `Rachel`) and a contextual explanation of the purpose of the line delivered in a male voice (which we refer to as `Andy`). The contextual explanation is created by sending each line, along with a carefully crafted prompt, to the OpenAI API. This uses the capabilities of LLMs, which are trained in an extensive corpus of text and code. The prompt is explicitly designed to guide the model to produce explanations that are context-aware, aiming to describe not just what the line does but also why it exists in the broader logic of the program.

Once both verbatim narration and contextual explanation are generated, the system uses Eleven Labs TTS API to convert them into audio. For each line of code i, two separate audio files are created: `rachel_i.mp3` for verbal narration and `andy_i.mp3` for contextual explanation. Eleven Labs was selected for its high-quality, lifelike voice synthesis, its support for voice customization, and its flexibility in terms of language and integration options. The platform allows developers to choose from a variety of voices or even clone specific ones, which enables the clear separation of narration types via distinct male and female voices. In addition, Eleven Labs actively promotes its use for accessibility-focused applications, which aligns well with the goals of this paper. The API also supports customization of speech parameters, such as speed and stability, which offers room for further tuning of the user experience, which we plan to explore in future work of this extension.

A significant technical challenge in this project was achieving reliable sequential playback of the two audio narrations for each line of code. Ensuring that `Andy`' contextual explanation consistently follows `Rachel`' verbal narration—without gaps or overlap—proved difficult due to the lack of reliable playback completion triggers in the development environment. To address this, we used the `ffmpeg` multimedia framework to concatenate the two separate audio files into a single synchronized file `output_i.mp3` for each line i. By combining both narration types into a single file, the extension simplifies playback logic: when a user selects a line of code, only one audio file needs to be played. By default, this file contains `Rachel`'verbatim narration followed by `Andy`' contextual explanation. However, BVI users can choose to hear only `Rachel` or `Andy`, depending on their preference. This guaranties smooth and immediate sequential playback, resulting in a better and more consistent user experience.

4 Rationale and Hypothesized Benefits

We hypothesize that employing two distinct voices: 1) one designated for reading verbatim code syntax and 2) for providing contextual explanations can significantly enhance code comprehension for BVI users compared to the conventional approach of using a single TTS voice for all output.

The underlying rationale for this hypothesis is based on principles of auditory scene analysis, which involves parsing and grouping sounds to form representations of distinct auditory objects [13]. By assigning different voices to the code and the explanation, the aim is to leverage this natural perceptual mechanism to help listeners more easily differentiate these two critical streams of information, potentially reducing ambiguity and cognitive effort. Our hypothesis is that the use of distinct female (verbatim code) and male (contextual explanation) voices improves auditory code comprehension for BVI users. This approach takes advantage of auditory scene analysis principles, where voice differences in pitch and timbre facilitate perceptual stream segregation [3, 8]. By perceptually separating the code syntax from its semantic explanation using distinct voices (male and female), our aim is to reduce the cognitive load associated with simultaneously processing syntax and semantics, similar to techniques used to separate real-world and virtual sounds in mixed-reality environments [5] Therefore, while the hypothesis is plausible and grounded in sound perceptual principles, its practical efficacy to improve code comprehension remains unproven. The potential cognitive benefits of easier stream segregation must be weighed against the potential costs of attention management, and this balance can only be determined through direct empirical testing on realistic coding tasks. In future studies, we hope to perform a randomized control trial experiment and validate the efficacy of our extension in VSCode.

5 Results

Table 1 presents a line-by-line comparison of how the selection sort algorithm, shown in Figure 1, is transcribed and narrated by three different systems: JAWS (based on our limited testing with the trial version), the verbatim `Rachel` voice, and the contextual Andy voice, both generated using ElevenLabs' TTS API. The actual audio recordings corresponding to Table 1, including playback of the voices of JAWS and `Rachel` and `Andy` , are available in an audio-only format [10]. We found that character-by-character playback from the JAWS trial version posed significant challenges for understanding code structure and logic, although we acknowledge that other configurations may be available in the full version.

Figure 1: Selection Sort algorithm displayed in VSCode. The top-right corner shows the `Narrate C++ Code` button, which triggers audio narration using `Rachel` and `Andy` voices. Once narration is complete for all code lines, a pop-up appears in the bottom-right corner with a green checkbox labeled `Narration Complete`.

While the `Rachel` voice provides a clear, line-by-line reading of the code syntax (e.g., "Line 3: void selectionSort open parenthesis..."), which is already an improvement over the character-by-character output observed from the JAWS trial version (e.g., "v o i d space s e l e c t i o n S o r t..."), the `Andy` voice demonstrates a deeper level of understanding.

The key advantage lies in the OpenAI-powered contextual narration provided by `Andy`, which synthesizes a high-level explanation of each code line rather than simply reading it aloud. This approach leverages the LLM capabilities of OpenAI to deliver concise summaries that preserve the programmer's intent As highlighted in Line 7, `Andy` does not read the line of code verbatim but mentions that Line 7 is an inner loop inside an outer loop and that the inner loop will iterate over the unsorted part of the array starting from $i + 1$. This semantic interpretation, derived from processing the code with an OpenAI model prompted for context-aware explanations, is something traditional screen readers typically cannot provide.

Further examples of this contextual understanding are evident throughout the table: Line 4: `Andy` explains the purpose of the outer loop within the selection sort algorithm ("placing the correct element at position i") and the reasoning behind the $n-1$ condition. Line 7: `Andy` clarifies the role of the inner loop ("examining every element in the unsorted portion") and its starting point relative to the outer loop $(i + 1)$. Lines 8 & 9: `Andy` explains the conditional logic being performed (checking for a smaller element) and the action taken if the condition is met (updating $minIndex$). Lines 15, 16, & 17: Each of these lines is in a closed curly bracket. `Andy` explains clearly that the first curly bracket closes the the if block. The second curly bracket closes the outer loop and the last curly bracket closes the selection sort function.

This demonstrates that the `Andy` voice, powered by OpenAI, captures and conveys the semantic meaning and structural context of the code, offering a significantly richer and more comprehensible auditory experience for users compared to the literal, non-interpretive playback characteristic of standard screen readers like the JAWS version tested.

Line #	Code Line	JAWS Transcription	Rachel Voice Transcription	Andy Voice Transcription
Line 1	#include <iostream>	# i n c l u d e < i o s t r e a m >	Line 1: #include less than iostream greater than	Line 1: This line includes the inputoutput stream library for Cplusplus.
Line 2	using namespace std;	u s i n g n a m e s p a c e s t d ;	Line 2: using namespace std semicolon	Line 2: Here, the code is indicating the usage of the standard namespace in Cplusplus.
Line 3	void selectionSort(int arr[], int n) {	v o i d s e l e c t i o n S o r t (i n t a r r [ ] , i n t n) {	Line 3: void selectionSort open parenthesis int arr open square bracket close square bracket int n close parenthesis open curly bracket	Line 3: This line defines a function named selectionSort that takes an integer array and its size as parameters.
Line 4	for (int i = 0; i < n - 1; ++i) {	f o r (i n t i = 0 ; i < n - 1 ; + + i) {	Line 4: for open parenthesis int i = 0 semicolon i less than n minus 1 semicolon plusplusi close parenthesis open curly bracket	Line 4: Inside the selectionSort function, there is a for loop that iterates over the array. This loop will run n minus 1 times.
Line 5	int minIndex = i;	i n t m i n I n d e x = i ;	Line 5: int minIndex = i semicolon	Line 5: Within the loop, an integer variable minIndex is declared and initialized with the value of i.
Line 6	// Find the index of the minimum element in the unsorted part	/ / F i n d t h e i n d e x o f t h e m i n i m u m e l e m e n t i n t h e u n s o r t e d p a r t	Line 6: comment Find the index of the minimum element in the unsorted part	Line 6: This is a comment describing the purpose of finding the index of the minimum element in the unsorted part.
Line 7	for (int j = i + 1; j < n; ++j) {	f o r (i n t j = i + 1 ; j < n ; + + j) {	Line 7: for open parenthesis int j = i plus 1 semicolon j less than n semicolon plusplusj close parenthesis open curly bracket	Line 7: Inside the outer loop, there is another nested loop that iterates over the unsorted part of the array starting from iplus1.
Line 8	if (arr[j] < arr[minIndex]) {	i f (a r r [j] < a r r [m i n I n d e x]) {	Line 8: if open parenthesis arr open square bracket j close square bracket less than arr open square bracket minIndex close square bracket close parenthesis open curly bracket	Line 8: This line checks if the element at index j is less than the element at the current minimum index to update the minimum index if needed.
Line 9	minIndex = j;	m i n I n d e x = j ;	Line 9: minIndex = j semicolon	Line 9: If the condition in line 8 is true, minIndex is updated to the current index j.
Line 10	}	}	Line 10: close curly bracket	Line 10: This is the end of the inner loop where the comparison is made to find the minimum element.
Line 11	}	}	Line 11: close curly bracket	Line 11: End of the inner loop block.
Line 12	// Swap the found minimum element with the first element	/ / S w a p t h e f o u n d m i n i m u m e l e m e n t w i t h t h e f i r s t e l e m e n t	Line 12: comment Swap the found minimum element with the first element	Line 12: This comment indicates the intention to swap the found minimum element with the element at the first index after the inner loop completes all iterations
Line 13	if (minIndex != i) {	i f (m i n I n d e x ! = i) {	Line 13: if open parenthesis minIndex negation = i close parenthesis open curly bracket	Line 13: Here, an if condition checks if the minIndex is not equal to i, meaning a smaller element was found, then it proceeds to swap the elements.
Line 14	swap(arr[i], arr[minIndex]);	s w a p (a r r [i] , a r r [m i n I n d e x]) ;	Line 14: swap open parenthesis arr open square bracket i close square bracket, arr open square bracket minIndex close square bracket close parenthesis semicolon	Line 14: Within the if condition, the swap function is called to interchange the elements at index i and minIndex.
Line 15	}	}	Line 15: close curly bracket	Line 15: End of the condition block checking for the swap condition.
Line 16	}	}	Line 16: close curly bracket	Line 16: End of the outer loop block that iterates through the array n minus 1 times to sort the elements.
Line 17	}	}	Line 17: close curly bracket	Line 17: End of the selectionSort function block.

Table 1: Comparison of Code Transcriptions by JAWS, Female Voice, and Male Voice (Lines 1–17) on the selection sort algorithm.

6 Conclusion

This paper presented an accessibility extension designed to enhance the comprehension of C++ code for BVI developers within the widely used VSCode environment. The system employs two distinct synthetic voices generated via ElevenLabs: a female voice (`Rachel`) provides a precise, verbatim reading of each code line, ensuring syntactic accuracy, while a male voice (`Andy`) delivers a context-aware explanation. Contextual narration is generated using the OpenAI API, which analyzes each line of code to infer semantic meaning and describe its role within the broader program structure. This dual-voice approach aims to offer a deeper level of understanding than traditional screen readers, which typically provide only literal readings. The use of distinct voices also supports auditory stream segregation, potentially reducing the cognitive load on the listener. The extension integrates inside VSCode, allowing BVI users to navigate code using the up and down arrow keys and trigger the synchronized playback of both the verbatim and contextual narrations for the selected line on demand. Together, these features form a cohesive framework for auditory code comprehension that bridges the gap between syntactic precision and semantic understanding. Future work will involve conducting user studies to evaluate the effectiveness of the tool in improving comprehension and efficiency, as well as exploring extensions to additional programming languages.

Acknowledgment

We gratefully acknowledge the use of OpenAI's ChatGPT for proofreading, grammatical checks, and other text editing tasks.

References

[1] Rafi Adnin and Maitraye Das. “"I look at it as the king of knowledge": How Blind People Use and Understand Generative AI Tools”. In: *Proceedings of the 26th ACM SIGACCESS Conference on Computers and Accessibility (ASSETS ’24)*. 2024. DOI: 10.1145/3663548.3675631.

[2] Khaled Albusays and Stephanie Ludi. “Eliciting programming challenges faced by developers with visual impairments: exploratory study”. In: *Proceedings of the 9th international workshop on cooperative and human aspects of software engineering*. 2016, pp. 82–85.

[3] Christian Brodbeck and Jonathan Z Simon. “Cortical tracking of voice pitch in the presence of multiple speakers depends on selective attention”. In: *Frontiers in Neuroscience* 16 (2022), p. 828546.

[4] Pierre Carbonnelle. *Top IDE Index*. Accessed: 2025-04-19. 2023. URL: https://pypl.github.io/IDE.html.

[5] Ruei-Che Chang et al. “SoundShift: Exploring Sound Manipulations for Accessible Mixed-Reality Awareness”. In: *Proceedings of the 2024 ACM Designing Interactive Systems Conference*. 2024, pp. 116–132.

[6] Ehtesham-Ul-Haque, Syed M. Monsur, and Shams M. Billah. “Grid-Coding: An Accessible, Efficient, and Structured Coding Paradigm for Blind and Low-Vision Programmers”. In: *Proceedings of the 35th ACM Symposium on User Interface Software and Technology (UIST ’22)*. ACM, 2022, 44:1–44:21. DOI: 10.1145/3526113.3545620.

[7] Carmen Flores-Saviaga et al. “The Impact of Generative AI Coding Assistants on Developers Who Are Visually Impaired”. In: *arXiv preprint arXiv:2503.16491* (2025). URL: https://arxiv.org/abs/2503.16491.

[8] João Guerreiro and Daniel Gonçalves. “Faster text-to-speeches: Enhancing blind people’s information scanning with faster concurrent speech”. In: *Proceedings of the 17th international ACM SIGACCESS conference on computers & accessibility*. 2015, pp. 3–11.

[9] Aboubakar Mountapmbeme, Obianuju Okafor, and Stephanie Ludi. “Addressing accessibility barriers in programming for people with visual impairments: A literature review”. In: *ACM Transactions on Accessible Computing (TACCESS)* 15.1 (2022), pp. 1–26.

[10] Prajwal Narayanaswamy. *Selection Sort Audio Demonstrations*. https://drive.google.com/drive/folders/1M63Xl-hAnTH4UpskM0dJEDGgcE1M9s2p usp=share_link. Audio files comparing JAWS screen reader and Rachel and Andy narrations for a C++ selection sort implementation. 2025. URL: https://drive.google.com/drive/folders/1M63Xl-hAnTH4UpskM0dJEDGgcE usp=share_link.

[11] Venkatesh Potluri et al. “CodeTalk: Improving Programming Environment Accessibility for Visually Impaired Developers”. In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 2018. DOI: 10.1145/3173574.3174192.

[12] Clark Saben, Jessica Zeitz, and Prashant Chandrasekar. “Enabling Blind and Low-Vision (BLV) Developers with LLM-Driven Code Debugging”. In: *Journal of Computing Sciences in Colleges* 40.3 (2024), pp. 204–215.

[13] Joel S Snyder, Claude Alain, and Terence W Picton. “Effects of attention on neuroelectric correlates of auditory stream segregation”. In: *Journal of cognitive neuroscience* 18.1 (2006), pp. 1–13.

[14] Stack Overflow. *2024 Developer Survey*. Accessed: 2025-04-19. 2024. URL: https://survey.stackoverflow.co/2024/.

[15] Andreas Stefik, Christopher Hundhausen, and Robert Patterson. “An empirical investigation into the design of auditory cues to enhance computer program comprehension”. In: *International Journal of Human-Computer Studies* 69.12 (2011), pp. 820–838.

[16] Naoto Takamatsu et al. "Computer Programming Education System for Visually Impaired Individuals". In: *Proceedings of the 14th International Workshop on Computer Science and Engineering (WCSE 2024)*. IEEE. 2024, pp. 64–69. DOI: 10.18178/wcse.2024.06.010.

[17] Sufyan Uzayr. *Mastering Visual Studio Code: A Beginner's Guide*. CRC Press, 2022.

[18] Erin Yepis. *Developers want more, more, more: the 2024 results from Stack Overflow's Annual Developer Survey*. Accessed: 2025-04-19. Jan. 2025. URL: https://stackoverflow.blog/2025/01/01/developers-want-more-more-more-the-2024-results-from-stack-overflow-s-annual-developer-survey/.

Intrusion Denied: Face Recognition on Raspberry Pi*

Metrine Nyaboke Osiemo[1], Dave Rushit[2], and Mansi Bhavsar[3]

[1]Department of Computer Information Science, Minnesota State University, Mankato, MN, USA, bokemetrine@gmail.com
[2]Department of Computer Information Science, Minnesota State University, Mankato, MN, USA, rushit.dave@mnsu.edu
[3]Department of Computer Information Science, Minnesota State University, Mankato, MN, USA, mansi.bhavsar@mnsu.edu

Abstract

With the growing concerns over personal and property security, there is a need for affordable and intelligent surveillance systems. Commercial options are often costly and cloud-dependent, limiting accessibility. This research presents a low-cost, AI-powered intrusion detection system using Raspberry Pi 5, featuring real-time facial recognition with Dlib and YOLOv4-Tiny. Designed for offline use, the system ensures privacy and efficiency on edge hardware. Performance comparisons across varied conditions revealed Dlib's higher accuracy in controlled environments, while YOLOv4-Tiny offered faster, more adaptable detection. This work provides a scalable, open-source solution for a secure, cost-effective home and small business surveillance.

1 INTRODUCTION

Intrusion detection systems have significantly evolved from traditional alarm-based mechanisms to advanced surveillance technologies that incorporate closed-circuit television (CCTV) and biometric systems. The integration of Artificial

Intelligence (AI) has further revolutionized surveillance by enabling real-time anomaly detection and behavior analysis, which are critical for ensuring security and prompt threat mitigation. Biometrics technologies, such as fingerprint and iris recognition, are now widely adopted as reliable methods for secure identity verification [1]. These advancements have shifted surveillance systems from passive monitoring to proactive security management, enhancing their effectiveness and responsiveness in real-world scenarios. AI-powered surveillance has gained prominence due to its ability to analyze vast amounts of data with high accuracy and minimal human intervention. Convolutional Neural Networks (CNNs) have proven highly effective for continuous monitoring in both business and residential settings, offering automated, real-time security solutions [2]. However, these systems often rely on computationally intensive models, which can be prohibitive for deployment in resource- constrained environments. This limitation highlights the need for optimized AI models capable of delivering safe security while operating efficiently on affordable hardware. By utilizing lightweight models, such as Mobile Nets, researchers have demonstrated that advanced AI functionalities can be adapted to devices like Raspberry Pi, providing practical, cost-effective surveillance solutions for households and small businesses [3]. Face detection and recognition on the other hand, are central components of modern surveillance systems. Algorithms such as the Viola-Jones algorithm have been widely adopted for their simplicity, accuracy, and low computational overhead, making them ideal for embedded platforms like the Raspberry Pi [4]. While advanced methods like Contextual Multi-Scale Region-based CNN (CMS-RCNN) can detect faces under challenging conditions (e.g., complex angles and lighting), these techniques require significant computational power, rendering them unsuitable for low-resource constrained environments [5]. By integrating efficient algorithms like Viola-Jones into this project, face detection can be achieved with minimal performance trade-offs, ensuring smooth and reliable operation on Raspberry Pi hardware Several research efforts have explored the application of machine learning algorithms for facial recognition in security systems. Systems employing Dlib, OpenCV, and YOLO models have demonstrated considerable promise for low-cost security. Raspberry Pi has emerged as a preferred platform for such edge applications due to its balance between processing power and cost. Literature reveals that real-time surveillance systems typically struggle with either speed or accuracy when deployed on lightweight hardware. Dlib has been favored for its strong face encoding capabilities using 128-dimensional vectors. It provides high accuracy but requires significant computational resources, which can slow down performance on edge devices. Conversely, YOLOv4-Tiny, a lightweight version of the YOLO family, trades some accuracy for faster frame processing and responsiveness. Research has validated YOLOv4-Tiny's effectiveness in dynamic environments with varying lighting and motion conditions, making it suitable for real-time surveillance tasks. For a comprehensive analysis of face recognition models optimized for low-power edge devices like Raspberry Pi, I refer to the study titled "IoT-MFaceNet: Internet-of- Things-Based Face Recognition Using MobileNetV2 and FaceNet Deep-Learning Implementations on a Raspberry Pi-400" [6]. This research evaluates the performance of MobileNetV2 and FaceNet models, focusing on metrics such as precision, recall, F1-score, and processing speed, providing valuable insights into balancing real-time performance and computational efficiency in embedded AI-driven security systems. Additionally, the study "Real-time Performance Comparison of Face Detection Algorithms on Raspberry Pi" [7] offers an in-depth evaluation of face detection techniques, including Haar Cascade, Dlib HOG, MTCNN, and MediaPipe, implemented on the Raspberry Pi. The research assesses these models based on accuracy, speed, and frames per second, highlighting practical challenges such as varying lighting conditions, facial expressions, occlusions, and poses, which are crucial for developers and researchers working on edge AI applications.

(**10**) single-spaced pages including tables, figures, and a list of references or bibliography.

2 SYSTEM DESIGN AND ARCHITECTURE

The architecture of this intrusion detection system is centered around the seamless integration of real-time video processing, deep learning models, and lightweight embedded computing. Designed to operate on Raspberry Pi 5, the system harnesses the capabilities of two separate facial recognition pipelines YOLOv4-Tiny for high-speed face detection and Dlib for facial feature encoding and recognition each running independently but governed by a unified user interface and execution logic. At the core of the system is a live video feed captured using the PiCamera2 module, which interfaces directly with the Raspberry Pi through the CSI port. Once activated, the camera continuously streams frames into the system. The user initiates the detection process via a Tkinter-based graphical user interface (GUI), which includes buttons for training, running recognition, capturing faces, and exiting the application. Once the "RUN TEST" button is pressed, the system enters a controlled sequence of operations executing the chosen model (either YOLO or Dlib) in a two-phase cycle: first running general detection for 5 seconds, followed by a 15-second face recognition window. After this interval, the model displays its precision and recall metrics directly on the GUI overlay. In the Dlib model pipeline, the system first detects faces in the frame using a Histogram of Oriented Gradients (HOG) or CNN-based detector. It then extracts 68 facial landmarks and encodes each detected face into a 128-dimensional vector a mathematical "fingerprint" of the face. These vectors are compared against a pre-trained local database using Euclidean distance. If the distance is below a defined threshold, a match is confirmed and the user's name is displayed; otherwise, the face is flagged as "Unknown." In contrast, the YOLOv4-Tiny model focuses on high-speed detection, employing deep learning to locate faces within frames rapidly. YOLO's strength lies in its efficiency, allowing it to detect and track multiple faces in real-time, even under limited hardware constraints. Once a face is detected, the system moves to identification, using its bounding box predictions to isolate and process candidate faces against the stored encodings. Like Dlib, YOLO outputs identification results that are used to compute precision and recall after each test cycle. To support smooth operation, the entire system is developed using Python chosen for its compatibility with OpenCV, Dlib, and deep learning frameworks. OpenCV handles the frame acquisition, image conversion, and bounding box overlays. The Tkinter GUI not only manages control logic but also displays real-time detection results, model scores, and allows users to rerun or terminate sessions easily. A local face database is maintained within the Raspberry Pi's file system, and sample face images (e.g., Metrine1.jpg, Robby1.jpg) are preloaded for testing. Additionally, an extended dataset for training or benchmarking is available on Kaggle, with instructions provided in the project's GitHub repository. Figure 1 and Figure 2 below are flowcharts showing how my proposed application system works.

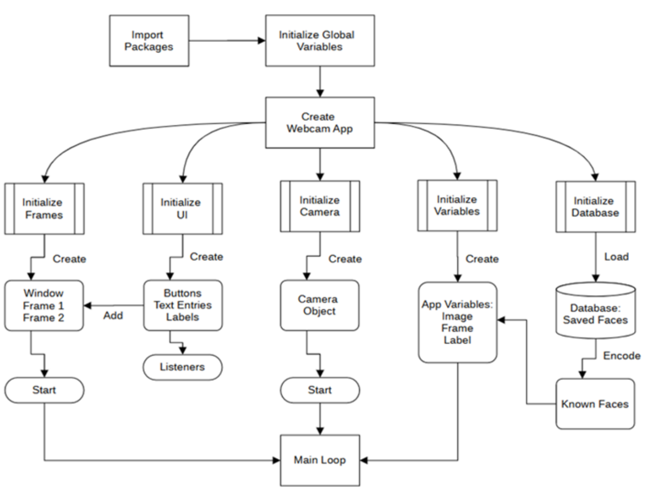


Figure 1: Overview of proposed system

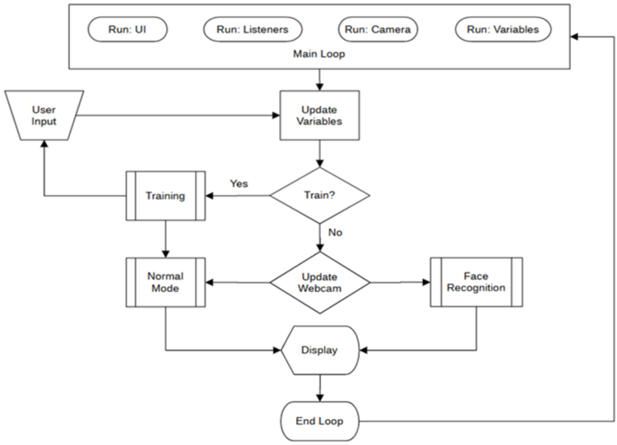


Figure 2: Overview of proposed system

3 RELATED WORKS

3.1 AI-Based Intrusion Detection Systems on Edge Devices (Raspberry Pi)

The development of on-device intrusion detection systems increasingly relies on efficient and robust face recognition algorithms that can function under real-world constraints such as variable lighting, pose variations, and limited computational resources. One of the most significant advancements in this field is the introduction of FaceNet, a deep learning-based approach for face recognition and clustering. [8] proposed Face Net as a unified embedding system that directly maps face images to a compact Euclidean space, where distances between embeddings correspond to facial similarity. Unlike traditional methods that rely on handcrafted features, FaceNet leverages deep convolutional neural networks (CNNs) trained with a triplet loss function, ensuring that embeddings of the same individual remain closer while embeddings of different individuals remain farther apart. Recent years have seen a surge of interest in AI-based intrusion detection systems (IDS) employing face recognition on resource-constrained edge devices such as Raspberry Pi, NVIDIA Jetson, and similar microcomputers. Researchers have explored a broad spectrum of machine learning and deep learning techniques ranging from traditional supervised methods like Support Vector Machines (SVM) to more advanced, unsupervised anomaly detection algorithms all aimed at automating the recognition of unauthorized individuals in real time [9].

3.2 Multi-Factor (Multimodal) Authentication of Edge Devices

Multi-factor authentication (MFA), also known as multimodal biometric verification, has emerged as a structured security layer, combining different biometric and contextual factors to reduce unauthorized access. In edge computing scenarios, including those using Raspberry Pi, MFA can integrate face recognition, voice authentication, and behavioral biometrics (such as keystroke or gait analysis) to create a comprehensive security framework. Recent studies have focused on making such MFA approaches computationally efficient and privacy-preserving, given the sensitivity of biometric data. For instance, [10] developed a voice-plus-face authentication system that integrates FaceNet for facial recognition and a Gaussian Mixture Model for voice recognition. Their approach, which employs score-level fusion, optimizes biometric authentication for resource-constrained devices, reducing latency through on-device inference while improving accuracy. The proposed multimodal biometric fusion scheme block diagram is shown in Figure 3 below.

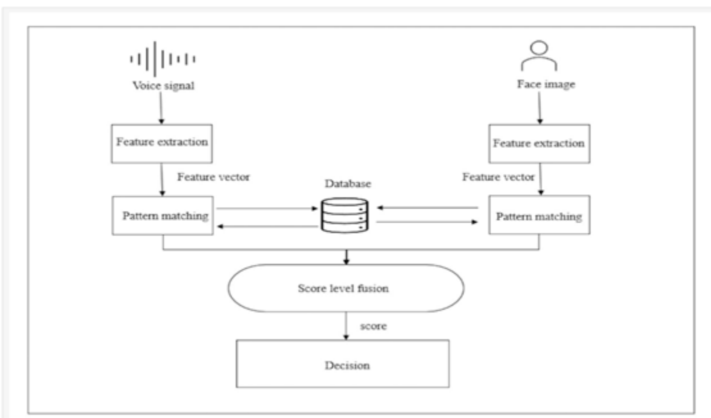


Figure 3: Multimodal biometric fusion scheme

3.3 Face Recognition

Face recognition technology has evolved significantly since its initial implementations in the 1960s, transitioning from rudimentary, feature-based methods to sophisticated machine learning and deep learning techniques [3]. Early approaches, such as the eigenfaces method, relied on principal component analysis to extract facial features, whereas modern systems leverage advanced algorithms capable of learning intricate patterns in facial data. These evolutions have made face recognition a key area of research in computer vision, offering robust performance in diverse conditions including variations in lighting, facial expression, and occlusions. With the emergence of small yet powerful devices like the Raspberry Pi, face recognition frameworks can now be deployed in low-cost, embedded environments. By integrating libraries such as OpenCV and Dlib, developers can implement real-time face detection and recognition on the Raspberry Pi for a variety of applications, including home security, automated attendance systems, and human-computer interaction. The minimal power consumption and portability of the Raspberry Pi make it ideal for projects that require on-device processing without relying on cloud resources.

3.4 YOLO-Based Face Detection

The YOLO (You Only Look Once) framework has become a cornerstone for object detection due to its ability to streamline the entire detection process within a single neural network. Initially introduced in 2016 by [11], subsequent iterations have continued to enhance both speed and precision. The YOLO model system detection can be seen in Fig 4 below. In the context of this project, YOLO is particularly valuable for rapidly localizing faces in real-time surveillance feeds. Its evolution from early versions to more advanced implementations like YOLOv7 illustrates how ongoing improvements in neural network design and compound scaling strategies can be leveraged to achieve high performance even on edge devices [12].

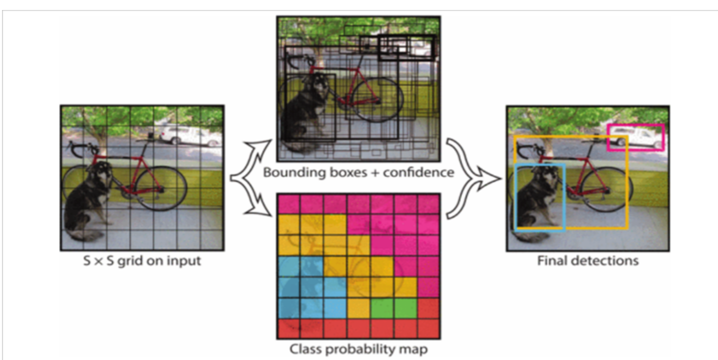


Figure 4: Yolo model as regression problem

3.5 Dlib Face Recognition

The Dlib library has emerged as a popular tool for face recognition due to its strong performance on benchmark datasets such as Labeled Faces in the Wild (LFW) [13]. The face recognition functionality provided by Dlib involves a complete pipeline that first detects faces using methods based on either CNNs or HOG, then predicts key facial landmarks, aligns the face, and finally extracts a 128-dimensional feature vector that characterizes the face [14]. In the experiments conducted, the Dlib-based face recognition system was evaluated for its performance in real-time scenarios. The findings align with those reported by [15], who conducted a comprehensive performance evaluation of face recognition software utilizing Dlib and OpenCV libraries. Their study highlighted Dlib's efficiency in facial recognition tasks, particularly noting its accuracy and computational performance. The figure 5 below illustrates the comparison of the results using OpenCV and Dlib libraries.

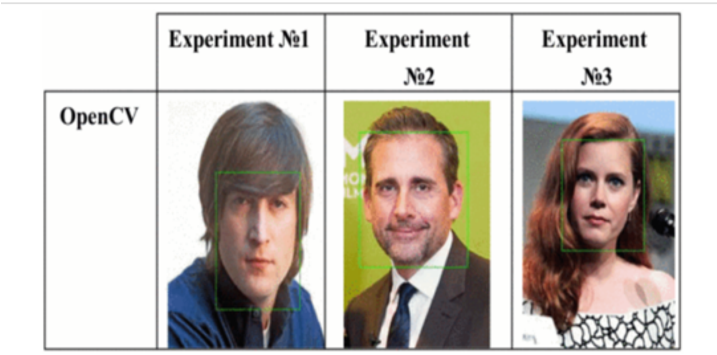


Figure 5: Comparison results from OpenCV libraries and Dlib

This vector-based representation allows for effective and efficient comparisons between faces, making Dlib a preferred choice in environments where computational resources are limited. For this project, which targets cost-effective and real-time intrusion detection, the lightweight yet powerful capabilities of Dlib ensure that face recognition can be executed reliably without the need for high-end hardware.

4 LIMITATIONS

While this system demonstrates the feasibility of deploying AI-powered surveillance on low-cost hardware, several limitations must be acknowledged. First, the computational constraints of the Raspberry Pi 5 limit the ability to scale to larger models or handle high-resolution processing. As a result, performance under multiple face detections or real-time processing at higher frame rates may be suboptimal. Secondly, both Dlib and YOLO models exhibited sensitivity to environmental conditions such as lighting, face orientation, and camera angle. These factors occasionally affected recognition accuracy, particularly in low-light or non-frontal views. Additionally, the system relies on a small, static facial database. Individuals who are not enrolled in the system is simply labeled as "unknown," which could result in false alarms or missed context in real-world applications. Lastly, while the system performs local inference efficiently, integration with real-time alert mechanisms and remote access was not implemented in this phase, leaving room for future enhancements. Despite these limitations, the project successfully showcases a functional, deployable surveillance prototype with practical applications in resource-constrained environments.

5 RESEARCH STATEMENT

Further research is needed to explore the long-term effectiveness and adaptability of low-cost Raspberry Pi–based facial recognition systems for home surveillance. It remains unclear whether lightweight models such as Dlib and YOLOv4-Tiny can maintain consistent performance across diverse real-world environments, especially under varying lighting conditions, face angles, and occlusions. Additionally, further investigation is required to assess the scalability of such systems in multi-user households and to determine whether edge-based solutions can match the precision of cloud-based alternatives without compromising privacy or computational efficiency.

6 CONCLUSION

This research validates that a cost-effective, AI-powered intrusion detection system can be successfully implemented on a Raspberry Pi platform. By combining the strengths of Dlib and YOLOv4-Tiny, the system offers a viable alternative to commercial surveillance solutions. It ensures real-time facial recognition, privacy preservation, and operational scalability, especially for underserved communities. Future work will be added to voice recognition, cloud-based alerts, and multi-camera integration to enhance coverage and functionality.

The full implementation of the Intrusion Detection System using Raspberry Pi, including both Dlib and YOLO models, is available on GitHub at the following link:

https://github.com/MetrineBokeh/ai-intrusion-detection-pi

References

[1] A. K. Jain, A. Ross, and S. Prabhakar, "An introduction to biometric recognition," *IEEE Transactions on circuits and systems for video technology*, vol. 14, no. 1, pp. 4–20, 2004.

[2] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[3] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," *ACM computing surveys (CSUR)*, vol. 35, no. 4, pp. 399–458, 2003.

[4] M. Sajjad, M. Nasir, K. Muhammad, S. Khan, Z. Jan, A. K. Sangaiah, M. Elhoseny, and S. W. Baik, "Raspberry pi assisted face recognition framework for enhanced law-enforcement services in smart cities," *Future Generation Computer Systems*, vol. 108, pp. 995–1007, 2020.

[5] C. Zhu, Y. Zheng, K. Luu, and M. Savvides, "Cms-rcnn: contextual multi-scale region-based cnn for unconstrained face detection," *Deep learning for biometrics*, pp. 57–79, 2017.

[6] A. S. Mohammad, T. G. Jarullah, M. T. Al-Kaltakchi, J. Alshehabi Al-Ani, and S. Dey, "Iot-mfacenet: Internet-of-things-based face recognition using mobilenetv2 and facenet deep-learning implementations on a raspberry pi-400," *Journal of Low Power Electronics and Applications*, vol. 14, no. 3, p. 46, 2024.

[7] R. Aswanandini and C. Deepa, "Comparison of advanced classification algorithms based intrusion detection from real-time dataset," *Automatic Control and Computer Sciences*, vol. 57, no. 3, pp. 287–295, 2023.

[8] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[9] W. Hao, R. Zhang, S. Li, J. Li, F. Li, S. Zhao, and W. Zhang, "Anomaly event detection in security surveillance using two-stream based model," *Security and Communication Networks*, vol. 2020, no. 1, p. 8876056, 2020.

[10] B. Alharbi and H. S. Alshanbari, "Face-voice based multimodal biometric authentication system via facenet and gmm," *PeerJ Computer Science*, vol. 9, p. e1468, 2023.

[11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[12] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 7464–7475.

[13] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database forstudying face recognition in unconstrained environments," in *Workshop on faces in'Real-Life'Images: detection, alignment, and recognition*, 2008.

[14] D. E. King, "Dlib-ml: A machine learning toolkit," *The Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, 2009.

[15] N. Boyko, O. Basystiuk, and N. Shakhovska, "Performance evaluation and comparison of software for face recognition, based on dlib and opencv library," in *2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP)*. IEEE, 2018, pp. 478–482.

AgriFedNet: Privacy-Preserving Plant Pathology with MobileNetV3-Small*

Ritesh Janga[1], Dr. Rushit Dave[1], and Mansi Bhavsar[1]
[1]Department of Computer Information Science
Minnesota State University
Mankato, MN 56001
ritesh.janga@mnsu.edu, rushit.dave@mnsu.edu, mansi.bhavsar@mnsu.edu

Abstract

Federated learning (FL) offers a privacy-preserving decentralized training model appropriate for plant disease classification across distributed sources. This manuscript constructs an FL model with a customized FedAvg algorithm and MobileNetV3-Small for plant disease classification in the 20,638-image PlantVillage dataset belonging to 15 categories. Three clients with severely imbalanced data distributions (80%, 15%, 5%) and overall updates in three rounds at the central server. The method keeps the best accuracy-based model and monitors loss, accuracy, precision, recall, and F1-score metrics. Federated model achieves 99.67% accuracy compared to local models (98.45%, 92.34%, 69.64% for clients 0, 1, and 2) and reduces global loss to 0.0115 by round 3, while local losses range from 0.0440–0.9262. Visualization, such as confusion matrices, displays the stability of the model across classes. These results demonstrate the potential of FL for large-scale, privacy-preserving agricultural diagnostics, data imbalance and resource constraint alleviation, with future focus on resource-constrained settings.

Keywords: Federated Learning, Plant Disease Classification, MobileNetV3-Small, PlantVillage, FedAvg

1 Introduction

Plant disease poses a worldwide food security risk through causing severe annual crop losses, necessitating early and proper diagnosis to reduce losses through timely intervention that protects farm production. Traditional machine learning of plant disease categorization is inclined to employ centralized data sets, consolidating information from farms into a single repository for training [7], but this is privacy-intrusive because farm-specific sensitive information, like images of infected crops, can be exposed, and it struggles to generalize over heterogeneous, diverse data distributions found in real-world agriculture [13]. Federated learning (FL) overcomes these issues using distributed collaborative training without sharing real data [8]. In FL, each client (e.g., a farm) trains a local model on private data, transmitting updates like weights to a server, which gets aggregated into a global model, without privacy loss while leveraging pooled knowledge [9]. This decentralized approach is suitable for agriculture, where information traverses geographically dispersed farms with varying disease and environmental statuses. This work uses FL to classify plant diseases on the PlantVillage dataset, a 20,638-image benchmark across 15 classes [5], with MobileNetV3-Small, a lightweight CNN for edge devices [4]. With a tailored FedAvg strategy [9], three clients (80%, 15%, 5% data), the federated model has an accuracy of 99.67%, which surpasses local models, as demonstrated through metrics and visualizations.

2 Literature Survey

Federated learning (FL) is a privacy-sensitive machine learning approach, with applications in a wide range of domains, such as agriculture. FedAvg algorithm computes client updates averaged by dataset size, with success when used for training deep networks on decentralized data [9]. Methods to reduce communication efficiency in FL attack building-block challenges in distributed computing [7]. Such contributions led to applying FL where data privacy is important, e.g., agriculture [8]. In agriculture, FL was used for predicting crop yields and identifying pests. Another used FL to predict yields across farms while preserving privacy, showcasing FL's ability to handle heterogeneous data distributions [3]. Another developed a privacy-preserving framework for pest detection using FL, surpassing local models with lightweight architectures [14]. These attest to FL's potential, albeit FL-based plant disease classification is less studied. The PlantVillage dataset is the gold standard for plant disease classification [5]. Deep learning on this dataset using centralized approaches was reported to have achieved high accuracies, one study achieving 99.35% accuracy with AlexNet but observing real-world variation problems such as variations in light [10]. Another was successful with ResNet and EfficientNet, pointing to the sensitivity of data augmentation to robustness [2]. Centralized approaches are, however, founded on the requirement for data aggregation, which becomes problematic when privacy is a concern. FL enables cooperative learning without the exchange of data. FL is advantageous to distributed analysis of agricultural data, with generalization improved over heterogeneous data, satisfying the requirements of this work [6]. Light models in FL minimize computational cost, with MobileNetV3-Small being an edge device-specific model, which fits our architecture choice [4]. FL breaks records that light models minimize overhead [12]. Recent work proposes adaptive aggregation techniques for heterogeneous FL optimization [11] and introduces the Flower framework, used here, for flexible FL systems [1]. These works point to FL's promise for privacy-preserving crop uses such as plant disease classification.

3 Methodology

3.1 Dataset

The PlantVillage dataset contains 20,638 RGB images of leaves from 15 classes, comprising healthy and diseased states of crops like tomatoes and potatoes [5]. The diseases covered in the classes are bacterial spot, early blight, late blight, leaf mold, and yellow leaf curl virus. To simulate the data imbalance in practical situations, the dataset is distributed among three clients in ratios of 80% (16,510 samples), 15% (3,095 samples), and 5% (1,033 samples). A test set globally, 20% of data (4,127 samples), is used for model performance validation.

3.2 Model Architecture

MobileNetV3-Small pre-trained on ImageNet is the foundation of this work [4]. The model is adapted to make computations more efficient as well as accommodate the PlantVillage dataset. Early feature layers (features[0:6]) are frozen in a bid to maintain general features acquired from ImageNet, and later feature layers (features[6:]) and the classifier are fine-tuned. The final linear layer in the classifier is modified to include 15 classes, which aligns with the label space of the dataset. This setup maintains the model relatively light, containing approximately 1.5 million parameters, which is suitable for edge devices.

3.3 Federated Learning Framework

The federated learning (FL) system includes a server and three clients and is conducted using the Flower framework [1]. The server uses a customized FedAvg strategy, SaveFedAvg, which sums client weights by data size per round, calculates global and local loss, accuracy, precision, recall, and F1-score metrics, saves the best model by global accuracy, and saves aggregated parameters but keeps histories of metrics for diagnostics. Each client, being a FlowerClient, undertakes local training for three epochs with the Adam optimizer using differential learning rates (1e-3 for classifier and 1e-4 for feature). In consideration of memory efficiency, mixed-precision training by PyTorch AMP reduces memory usage and speeds up training. Data augmentation like random horizontal flipping of the image, rotation of the image, and color jittering is done to improve the model's robustness to the variation of the leaf image so that it can perform well in generalizing for plant disease prediction under different conditions. The application runs on a 16GB RAM, NVIDIA RTX 3080 GPU server and requires a stable internet network with a minimum bandwidth of 10 Mbps and less than 50ms latency for smooth communication between clients and servers.

3.4 Training Process

The training process has three rounds of federated learning. In every round, the clients train local models on their own data and send updated weights to the server. The server aggregates these weights with FedAvg and evaluates the global model on the test set. For comparison, the local models are trained separately on each client's data using the same hyperparameters (three epochs, Adam optimizer, and data augmentation) on a local host with 8GB RAM and an Intel i5 processor, supplemented by a network bandwidth of 5 Mbps and periodic 4G fallback to simulate rural conditions. Federated and local approaches are validated per round so performance improvement can be monitored.

3.5 Evaluation Metrics

Performance is quantified by a range of metrics computed on the global test set, such as federated (FedAvg) as well as local models to enable direct comparison and mark benefits of collaborative learning. These measures include cross-entropy loss, indicating the model prediction's error; accuracy, representing the proportion of correct predictions over all classes; precision, macro-averaged proportion of true positives to positive predictions; recall, macro-averaged proportion of true positives to positive instances; and F1-score, macro-averaged harmonic mean of precision and recall. By assessing these metrics, the test provides an overall view of how the model performs in plant disease classification and reflects the benefits of the federated approach compared to local training.

4 Results

4.1 Performance Metrics

Table 1 shows the resulting statistics after three iterations between the federated (FedAvg) model and locally averaged models across all clients. The federated model has a loss of 0.0115 and an accuracy rate of 99.67%, which is way above the local models with an average loss of 0.3987 and an accuracy rate of 86.81%.

Table 1: Final Metrics Comparison (Round 3)

Metric	FedAvg (Round 3)	Local	Difference (FedAvg-Local)
Loss	0.0115	0.3987	-0.3873
Accuracy	0.9967	0.8681	0.1286
Precision	0.9971	0.8935	0.1036
Recall	0.9956	0.8619	0.1337
F1	0.9963	0.8602	0.1361

This table indicates the federated (FedAvg) and local model averaged metrics for loss, accuracy, precision, recall, and F1-score, and their differences. Negative loss difference and positive differences in other metrics reflect FL's improved performance.

4.2 Client-Specific Results

The client-specific results indicate the robustness of federated learning across various data distributions. For Client 0, which has 80% of the data, the local model achieves an accuracy of 98.45% and a loss of 0.0440, but the FedAvg testing improves this to an accuracy of 99.85% with a reduced loss of 0.0063 by round 3, indicating the advantage of collaborative learning even for the client with the largest set of data. Client 1 with 15% of the data sees its local model reach accuracy at 92.34% and loss of 0.2261, which FedAvg improves to 99.64% accuracy and a loss of 0.0113 from other clients' work. Meanwhile, Client 2 with a mere 5% of the data only has a local model accuracy of 69.64% and a high loss of 0.9262 due to sparse data; FedAvg improves its performance, however, significantly to an accuracy of 99.52% and a loss of 0.0167, demonstrating the capability of federated learning to counter limitations arising from sparse data.

4.3 Metric Trends

Figures 1-15 illustrate metric trends across three rounds for Clients 0, 1, and 2, respectively, comparing FedAvg and local performance for accuracy, loss, precision, recall, and F1-score.

4.3.1 Client 0 Metric Trends

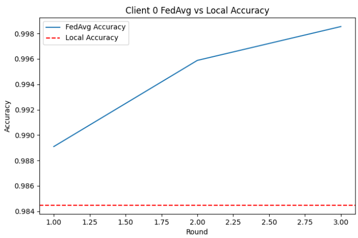


Figure 1: Client 0 Accuracy

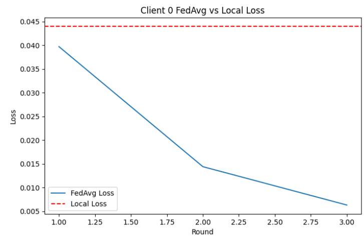


Figure 2: Client 0 Loss

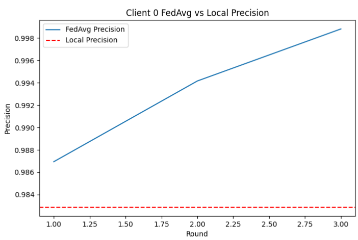


Figure 3: Client 0 Precision

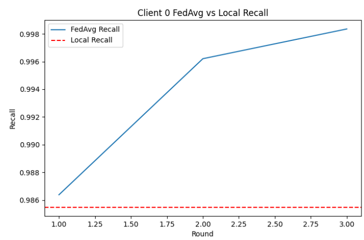


Figure 4: Client 0 Recall

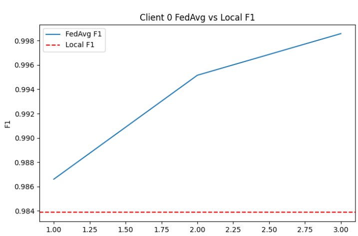


Figure 5: Client 0 F1 Score

The figures for Client 0 include five line plots comparing FedAvg (solid blue lines) and local (dashed red lines) metrics over three rounds. Accuracy increases from 0.99 to 0.998 (local: 0.985), loss decreases from 0.04 to 0.01 (local: 0.044), precision rises from 0.99 to 0.998 (local: 0.983), recall improves from 0.99 to 0.998 (local: 0.986), and F1-score grows from 0.99 to 0.998 (local: 0.984).

4.3.2 Client 1 Metric Trends

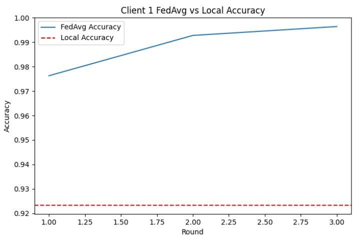


Figure 6: Client 1 Accuracy

Figure 7: Client 1 Loss

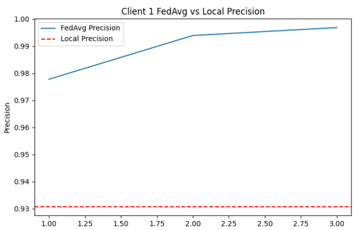


Figure 8: Client 1 Precision

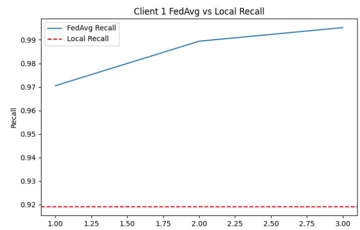


Figure 9: Client 1 Recall

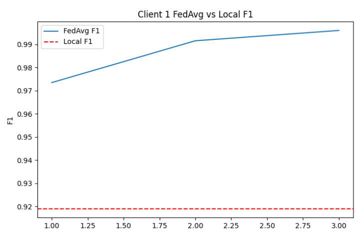


Figure 10: Client 1 F1 Score

The figures for Client 1 include five line plots comparing FedAvg (solid blue lines) and local (dashed red lines) metrics over three rounds. Accuracy increases from 0.97 to 0.99 (local: 0.923), loss decreases from 0.07 to 0.01 (local: 0.226), precision rises from 0.97 to 0.99 (local: 0.93), recall improves from 0.97 to 0.99 (local: 0.92), and F1-score grows from 0.97 to 0.99 (local: 0.92).

4.3.3 Client 2 Metric Trends

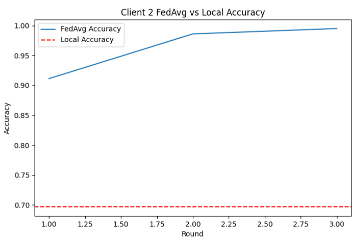


Figure 11: Client 2 Accuracy

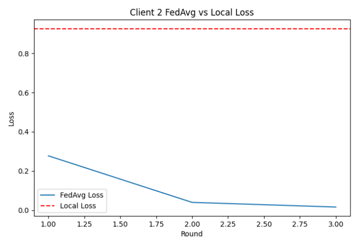


Figure 12: Client 2 Loss

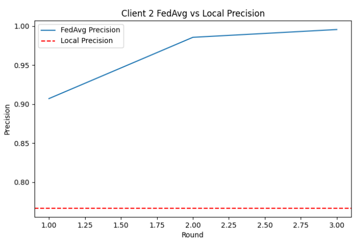


Figure 13: Client 2 Precision

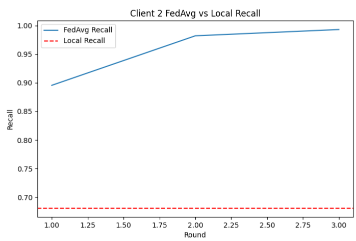


Figure 14: Client 2 Recall

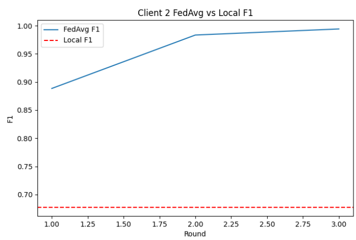


Figure 15: Client 2 F1 Score

The figures for Client 2 include five line plots comparing FedAvg (solid blue lines) and local (dashed red lines) metrics over three rounds. Accuracy increases from 0.90 to 0.99 (local: 0.696), loss decreases from 0.30 to 0.02 (local: 0.926), precision rises from 0.90 to 0.99 (local: 0.70), recall improves from 0.90 to 0.99 (local: 0.70), and F1-score grows from 0.90 to 0.99 (local: 0.70).

4.4 Confusion Matrices

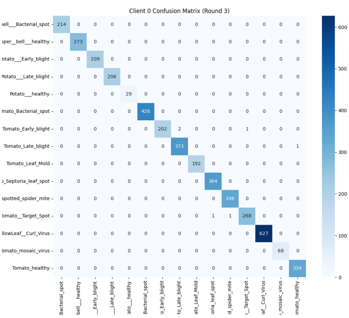


Figure 16: Client 0 Confusion Matrix (Round 3)

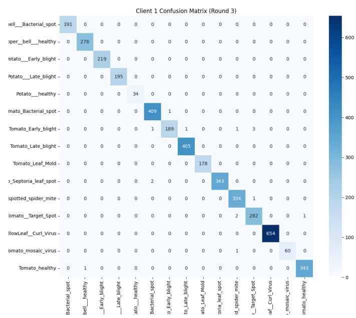


Figure 17: Client 1 Confusion Matrix (Round 3)

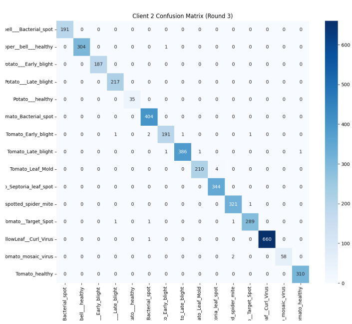


Figure 18: Client 2 Confusion Matrix (Round 3)

5 Discussion

Results validate federated learning's effectiveness in plant disease prediction, where federated model performance of 99.67% surpasses that of local models' 69.64% to 98.45%. Client 2 with 5% data sees most improvement—local model's 69.64% accuracy improves to 99.52% with Clients 0 and 1 updates, echoing FL's capability to mitigate data sparsity by cooperation in alignment with literature in FL over heterogeneous environments [6]. The FedAvg method, optimized for application, offers enhanced user-friendliness through the maintenance of the top-performing model for edge-device deployment, which helps farmers in disease identification. MobileNetV3-Small's 1.5 million-parameter model is suitable for environments with limited resources [4]. The 80%, 15%, and 5% data split is realistic to real-farm diversity, and the federated model's robustness reflects research that states FL enhances distributed agricultural data generalization [6]. Optimization is suggested, though, by the 25,063-second runtime, possibly by model compression or quantization [12]. Future work will require to remediate rural connectivity, confirm real-world variability of data outside the PlantVillage dataset [10], explore adaptive aggregation [11], privacy preservation via methods like differential privacy or secure multi-party computation [12], and scaling to more clients.

6 Conclusion

This research offers a federated framework of plant disease classification using MobileNetV3-Small on the PlantVillage dataset with a global accuracy of 99.67%. The new FedAvg algorithm has a scalable privacy-protecting solution for crop diagnosis using the combination of client updates, observation tracking, and model best storage. Particularly for low-data clients, the federated model performs better than local models to show FL's capacity to enhance generalisation over a range of datasets. Confusion matrices and metric graphs are a few of the visual tools that show their reliability across clients. The usage fee and use of controlled data of the framework indicate directions of future improvement including cost reduction, use of real-world data, and investigation into adaptive aggregation strategies to benefit FL in agriculture.

References

[1] Daniel J Beutel et al. "Flower: A friendly federated learning research framework". In: *arXiv preprint arXiv:2007.14390* (2020).

[2] Mohammed Brahimi, Kamel Boukhalfa, and Abdelouahab Moussaoui. "Deep learning for tomato diseases: Classification and symptoms visualization". In: *Applied Artificial Intelligence* 31.4 (2017), pp. 299–315.

[3] Vani Hiremani et al. "Federated learning for crop yield prediction: A comprehensive review of techniques and applications". In: *MethodsX* (2025), p. 103408.

[4] Andrew Howard et al. "Efficient Mobile Vision Transformers for Edge Devices". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE. 2020, pp. 12345–12354.

[5] David Hughes, Marcel Salathé, et al. "An open access repository of images on plant health to enable the development of mobile disease diagnostics". In: *arXiv preprint arXiv:1511.08060* (2015).

[6] Nyoman Karna et al. "Towards Precision Agriculture Using Federated Learning-Driven Crop Recommendation System". In: *2024 15th International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE. 2024, pp. 1538–1542.

[7] Jakub Konečnỳ et al. "Federated learning: Strategies for improving communication efficiency". In: *arXiv preprint arXiv:1610.05492* (2016).

[8] Tian Li et al. "Federated learning: Challenges, methods, and future directions". In: *IEEE Signal Processing Magazine* 37.3 (2020), pp. 50–60.

[9] Brendan McMahan et al. "Communication-efficient learning of deep networks from decentralized data". In: *Artificial Intelligence and Statistics*. PMLR. 2017, pp. 1273–1282.

[10] Sharada P Mohanty, David P Hughes, and Marcel Salathé. "Using deep learning for image-based plant disease detection". In: *Frontiers in plant science* 7 (2016), p. 215232.

[11] Sashank Reddi et al. "Adaptive federated optimization". In: *arXiv preprint arXiv:2003.00295* (2020).

[12] Tingting Wu, Chunhe Song, and Peng Zeng. "Efficient federated learning on resource-constrained edge devices based on model pruning". In: *Complex & Intelligent Systems* 9.6 (2023), pp. 6999–7013.

[13] Qiang Yang et al. "Federated machine learning: Concept and applications". In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 10.2 (2019), pp. 1–19.

[14] Junyong Yoon et al. "Privacy-Preserving Pest Detection Using Personalized Federated Learning". In: *International Conference on Agriculture-Centric Computation*. Springer. 2023, pp. 58–70.

A Case Study of Achieving Original Results through Independent Student Research*

Brendan A. De Leeuw, Jay Bagga
Department of Computer Science
Ball State University
Muncie, IN 47306
{brendan.deleeuw,jbagga}@bsu.edu

Abstract

In this paper, we describe the successful development and completion of research projects by students who enrolled in a 15-week online asynchronous class. This class, Applications of Computability, is part of the MS Program in Computer Science at Ball State. A main requirement of this class is the completion of a research project and a final research report by each student. A novel aspect of research activities is the peer review process. This asynchronous online class, with 38 students enrolled, was recently developed and taught by the faculty coauthor (Jay Bagga). The paper also includes a case study of original research completed by the student coauthor.

1 Introduction

In this paper we present a case study that shows the successful development and completion of research projects by students who enrolled in a 15-week asynchronous class. The admission to the MS Program in Computer Science at Ball State University is performance-based. Students who have completed an undergraduate degree in any discipline can apply for admission. Upon completing three 3-credit courses and earning a cumulative 3.0 GPA or better, a student is fully admitted to the program.

Several case studies have appeared in the literature that show the importance of student research projects at both undergraduate and graduate levels in computer science and related fields. We mention four research articles that have appeared in the Journal of Computing Sciences in Colleges. Fife [5] shows that authentic learning happens through student research projects at both undergraduate and graduate levels. The paper by Darby [4] describes a course that teaches the best practices for research methodology and data handling. Koeller [6] discusses strategies for successful student research, and Citron, Gomberg, and Seng [1] provide a case study where a design project for an engineering class was brought to market and describes educational outcomes achieved during the entire process.

It is a requirement of the Graduate School at our institution that students enrolled in an MS degree successfully complete six hours of work either through a thesis or courses that include a specific research component. Our case study concerns the class "Applications of Computability", which is one of a collection of classes that satisfy the research requirement. The students in this program come from a variety of backgrounds and include those who may not have had experience in doing research. Our class introduces and implements certain novel features which are described in the following sections.

The learning outcomes of this class include two outcomes that specifically address the research requirement.

- Formulate a research project to investigate and solve one or more problems. Design and develop a Research Project. Report findings in a Research Report in a prescribed format.
- Explain the scientific method for conducting research. Understand and use LaTeX as a tool for producing research reports and papers that include text, formulas and equations, images, charts, tables, citations, and references.

Through reading and videos, students learn about the research process, formulating a proposal, conducting a literature review, using Ball State library databases such as the ACM Digital Library, and other resources such as interlibrary loan, collecting data and reporting research using LaTeX.

Research assignments account for 52% of the grade in the class, with 48% credit coming from tests (quizzes and exams). Research activities include the completion of two reflective essays, the selection of a research topic, the formulation of a research proposal, a literature survey, the completion of the research project, and a final report describing the research. It is required that all students use LaTeX as the tool for all documentation, including the final research report.

In the following sections, we describe the research activities performed by students, present a case study detailing the research activities of one of the coauthors, and describe the specific research problem solved by this coauthor.

2 Research Activities

The topics of this class include the basic concepts of languages, finite-state machines, Turing machines, and various types of grammar with applications to parsing and compilers. The emphasis is on analyzing applications of computability to programming languages, networks, and cybersecurity. As they study and work with the content of the class, students are provided with a structured timeline for their research activities.

2.1 The Peer Review Process

Peer reviews are an integral part of the process. Each submission is peer-reviewed by two other students in the same class who provide feedback, which is then used to revise the reports submitted to the instructor. Each student receives peer reviews and also acts as peer reviewer for some other students.

The peer review is made up of a series of prompts that the reviewer is required to respond to. After initially checking if a file was loaded in the correct format (LaTeX or pdf), and has the required elements (such as references and citations), the reviewer grades the contents of the submitted report (based on a provided rubric). We include two examples of such prompts: (1) Does the report have an Introduction section with the definition of the problem, all relevant concepts/definitions needed to describe the problem, and describes the goals for a solution? (2) Rate the section that describes data collection and methodology. The reviewers are also encouraged to provide freeform responses to the above, although it does not effect the grades.

2.2 The Cadence of Research Activities

The following list provides a detailed list of the research activities. Table 1 shows the timeline of the activities. The two-week interval to revise the reports based on feedback from peer reviews worked well.

- Peer reviews of each student by two other students in the class.
- Introduction to LaTeX as a tool for creating and producing all reports in the class, and experimenting with basic and some advanced features of LaTeX and other packages such as $TikZ$.
- Introduction to the scientific method for conducting research
- Developing a research project on a topic related to Applications of Computability, getting it peer reviewed, revising it with the feedback received from peer-reviews, and getting it approved by the instructor.

- Conducting a survey of the literature and completing the project by using the proposed methodology.
- Preparing a draft report in a prescribed format which is peer reviewed and then revised with feedback from the peer reviews.
- Producing and submitting the Final Report which is then reviewed by the instructor.

Week due	**Assignment**	**Reviewer**
2	Experiment with LaTeX and submit an essay	Peer
6	Submit a draft of your Research Proposal	Peer
8	Submit the final Research Proposal	Instructor
10	Submit an essay on a topic of your choice	Peer
12	Submit a draft Research Report of your project	Peer
14	Submit the final Research of your project	Instructor

Table 1: Cadence of Research Activities

2.3 Discussion

Thirty-eight students proposed and successfully completed research projects on a wide variety of topics related to the theme of the course. Topics included applications to compilers, cybersecurity, network algorithms, and cloud computing. All students received 80% or higher in their research projects. The peer review feedback process generally worked very well, with some delays that resulted in students having extra time to submit their report.

Many of the students started with little or no experience with LaTeX. The reports showed that all students were able to master the basic and some advanced features of LaTeX quickly. Templates were provided for the Research Proposal and the Research Report. Students were required to conduct literature surveys.

In the next section, we provide a case study which describes research activities of the student coauthor. This coauthor introduced a new class of automata and showed that the set of languages that are accepted is between two other automata classes. We plan to submit the results obtained in a separate technical paper in an appropriate journal.

3 Case Study

The following is a overview of the student coauthor's research efforts over the course of the class.

3.1 Selection of a Research Topic

The students were first asked to start formulating some research topics using the textbook for inspiration or drawing from their other interests. This student coauthor wrote down six one-to-two sentence topic ideas, half of them application focused and the other half based on theory.

One of the goals of the research in the class is to investigate and solve a problem. This led the student coauthor to narrow down the topics to those that had concrete problems first. Students also wanted to select a topic they could make immediate progress on with the knowledge already gained in the class.

The initial idea the student coauthor started with was determining if any Pushdown automaton can be broken down into some kind of "composition" of Finite State Machines. An investigation of the language $L = \{a^n b^n : n > 0\}$ (a canonical example of a language requiring a Pushdown automaton (Rich [9])) showed that a finite state transducer "composed" with a 2-tape finite state machine could accept the language.

From that example we generalized to a 2-tape finite state machine where one tape would have the input word and the other would have the output of a transducer. The details of how this machine works were not defined at this point, but it gave the student coauthor a place to start his research. We were led to this question: what research exists on 2-tape finite state machines and transducers?

3.2 Initial Literature Gathering

In the first week, the class was introduced to the university's library database, journals, and other publication resources. Searching for 2-tape automata in the university's catalog and Google Scholar we discovered some initial papers. These papers expand on the 2-tape automata but did not serve as a good introduction. The student coauthor instead followed the citations in the papers. He found several papers and survey reports on 2-tape automata. Further searches on Google scholar yielded additional papers. This helped us get a broad view of the existing literature without going into the details.

During this search we also learned about 2-way and 2-head automata, along with the generalization to k-tape and k-head automata. While not immediately relevant, the 2-head automata turned out to be more relevant as the research progressed.

At this stage the student coauthor only skimmed the papers to see if any of the definitions or theorems seemed related. Short summaries for each paper were created to determine their relevance to the project. This helped the student coauthor to create a specific research question and provided further motivation. In particular, the paper by Nagy [8] showed that a variation of 2-tape automata accept exactly the linear context-free languages, a language class between the regular and context-free languages. We at first believed that all automata accepted exactly one of the four classes in the Chomsky Hierarchy but now knew that was not the case. This led to a specific research question: "What is the class of languages accepted by a 2-tape Finite State Machines when the second tape is the output of a Finite State Transducer run on the first tape? Was it a classic Chomsky hierarchy type or something else?"

3.3 Project Proposal

At this point the student co-author was using an unstated assumption that the transducer output at most one character at a time. While trying to formalize the automata for the proposal he realized this and saw it could be simplified to a read only 2-tape Finite State Machine with the same input on both tapes. From a study of our literature we knew this could be further simplified to a 2-head automata where only one head reads at a time.

We then focused the literature search on 2-head automata and found the Watson-Crick automata. These have a double stranded input, similar to a 2-tape machine, but the elements of the strands are linked in a relation (see Czeizler and Czeizler [2]). Kuske and Weigel [7] showed that requiring the two stands to have the same string on them does not reduce the class of languages accepted by Watson-Crick automata. Much like the automata we were using, having two tapes or one did not matter, so the Watson-Crick automata seemed to be closely related.

At this point we could not say for certain if these automata were something that has been studied before or a new concept. However, it seemed like the existing 2-head automata from the literature survey were different. We wrote the project proposal to study these automata further. Our main goal was to find proofs showing how this automata class relates to other classes, and how it fits into the Chomsky hierarchy.

3.4 Carrying out the Research

With the proposal approved, the student coauthor started to study some of the papers more thoroughly. This helped us describe exactly how previously studied multi-head automata were different from the one in the proposal. We further refined the definition to include other unstated properties that were implicitly assumed earlier.

We showed the automata could accept some context-free and context-sensitive languages by constructing examples that accepted $L = \{a^n b^n : n > 0\}$ and $L' = \{a^n b^n c^n : n > 0\}$. We gave them the name "1A-FSM(2)," since they have two heads and one of them is *active*, in the sense that it reads from input. We further defined other common automata terms such as *configuration*, *computation*, and *acceptance*, and started putting together sketches of proofs relating the 1A-FSM(2) to other automata classes.

We initially thought the 1A-FSM(2) and Watson-Crick automata accept the same class of languages. The first proof sketches were to show that equivalence. However, we could not resolve the determinism of 1A-FSM(2) automata with the non-determinism of Watson-Crick automata. This lead us to return to searching for literature. We found that Czeizler et al. [3] had introduced determinism to Watson-Crick automata. With these deterministic Watson-Crick automata we could complete the sketch proofs and submit the draft for student peer review.

3.5 Peer Review and Feedback

The peer feedback from two students was valuable in that it led us to make some structural changes to the paper. In the original draft it was not clear to the peers what the project's goals were and how they were achieved. We addressed this by reworking the sections so that all the goals and methodology were placed under one section instead of being scattered in different parts of the paper in its earlier version. The peer feedback also helped us clearly define how the rest of the paper would proceed and what the reader should expect to see.

In addition to receiving feedback, the student coauthor provided peer review to other students, and this helped us improve our own report. Our earlier draft did not have an abstract but after reading abstracts from other peers we decided to include one. Furthermore, we learned how to use `biblatex` through our own peer reviews of other students' essays. This motivated us to use this efficient method to manage citations in our paper.

3.6 Finishing Proofs and Final Submission

After addressing the feedback from the peer reviews we started to fill out the proof sketches and found an error in one of them. The proof had not shown that all languages accepted by the 1A-FSM(2) automata were accepted by deterministic Watson-Crick automata. That proof was removed and instead a weaker proof that all 1A-FSM(2) languages can be accepted by non-deterministic Watson-Crick automata was added.

The final change we made was adding an end marker to the input, allowing for the two heads to sense when they reached the end of the input. We noted that most of the proofs augmented the input to have an end marker and that the proofs that did not could be modified to also work with an end marker.

Letting the two heads sense the end of the string made it easier to compare languages accepted by other automata.

3.7 Research Results

The following briefly restates the original work presented in the final version of the student coauthor's report. The research paper introduced a new class of automata. These are the 1A-FSM(2) as defined below.

Definition 3.1 (1-limited 2-Head Finite State Machine (1A-FSM(2)))**.** A 1A-FSM(2) is a 6-tuple $M = (K_1, K_2, \Sigma, \delta, s, A)$ where the input is augmented by appending a single end of string character \$ and:

- Σ is a finite alphabet.
- K_1 and K_2 are finite sets of states where $K_1 \cap K_2 = \varnothing$.
  K_1 are the states associated with the reading from the first head, and K_2 the second.
- $\delta : K_1 \cup K_2 \times \Sigma \cup \{\$\} \to K_1 \cup K_2$
  The transition function is aware of the end of the string character.
- s is the start state, where $s \in K_1 \cup K_2$.
- A is the set of accepting states, where $A \subseteq K_1 \cup K_2$.

Figure 1 shows an example 1A-FSM(2) automaton to accept the language $\{a^n b^n : n > 0\}$. Reading the character on a transition advances the associated head one position. If there is no transition out of a state for the character read the automaton immediately rejects the string.

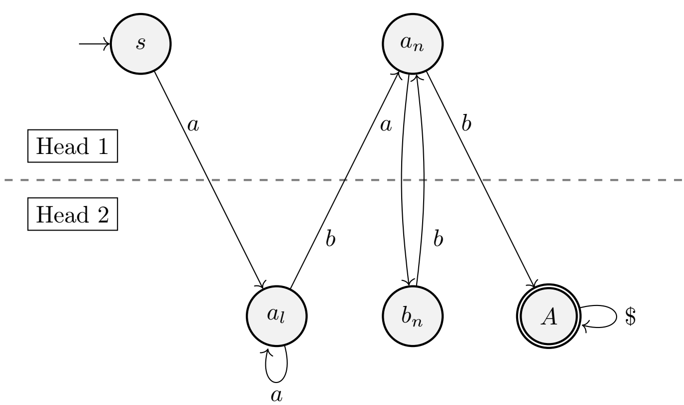


Figure 1: A 1A-FSM(2) to accept the language $\{a^n b^n : n > 0\}$.

The primary question that is investigated is: "What class of languages do these automata accept?" We showed they accept at least the languages accepted by the Strongly Deterministic Watson-Crick automata and at most the Non-Deterministic Watson-Crick automata. We posed future research on this class of automata by generalizing to automata with more than two heads. In particular, do they have a similar hierarchy to what Yao and Rivest [10] showed for multi-head automata when the number of active heads is fixed or allowed to vary?

4 Conclusion

In an asynchronous distance learning setting the class structure helped guide the student coauthor to conduct research and achieve new results, which we seek to publish in a technical paper in an appropriate journal. This case study illustrates that the class structure and pedagogical methodology trains students well on how to conduct independent research, do literature search and use tools such as LaTeX to report results. It further motivates students to achieve new results and to disseminate those in appropriate venues.

The peer review process is an important and integral part of the structure of this class. The feedback given and received through peer reviews leads to further learning and improvement of the research results and reports.

References

[1] Connor Citron, Brian Gomberg, and John Seng. "The Aithon board: a case study in commercialization of a student project". In: *J. Comput. Sci. Coll.* 29.4 (Apr. 2014), pp. 202–209. ISSN: 1937-4771.

[2] Elena Czeizler and Eugen Czeizler. "A Short Survey on Watson-Crick Automata." In: *Bull. EATCS* 88.3 (2006), pp. 104–119.

[3] Elena Czeizler et al. "On the descriptional complexity of Watson–Crick automata". In: *Theoretical Computer Science* 410.35 (2009), pp. 3250–3260.

[4] Miranda Malouf Darby. "Teaching Bioinformatics Students to Lead Reproducible Research". In: *J. Comput. Sci. Coll.* 40.3 (Oct. 2024), pp. 334–346. ISSN: 1937-4771.

[5] Leslie D. Fife. "Authentic learning through student research projects". In: *J. Comput. Sci. Coll.* 25.4 (Apr. 2010), pp. 37–43. ISSN: 1937-4771.

[6] Andreas Koeller. "Experiences with student research at a primarily undergraduate institution". In: *J. Comput. Sci. Coll.* 20.3 (Feb. 2005), pp. 181–187. ISSN: 1937-4771.

[7] Dietrich Kuske and Peter Weigel. "The role of the complementarity relation in Watson-Crick automata and sticker systems". In: *International Conference on Developments in Language Theory*. Springer. 2004, pp. 272–283.

[8] Benedek Nagy. "A class of 2-head finite automata for linear languages". In: *Triangle* 8 (2012), pp. 89–99.

[9] Elain Rich. *Automata, Computability and Complexity: Theory and Application*. Pearson Education, Inc., 2019.

[10] Andrew C. Yao and Ronald L Rivest. "$k + 1$ heads are better than k". In: *Journal of the ACM (JACM)* 25.2 (1978), pp. 337–340.

Development of Learning Outcomes Using Classroom Project Constraint Systems *

Travis Faas
Department of Computer Science
Ball State University
Muncie, IN 47306
travis.faas@bsu.edu

Abstract

Instructors often employ project constraints to ensure desired student learning outcomes. These constraints generate both constraint-related skills and knowledge of options discarded to satisfy constraints. In this paper I describe how I use constraints in game programming courses and use this discussion to suggest three categories of project constraints: direction, ideation, and execution. These constraints categories create opportunities for learning by encouraging students to explore several possible solutions during project design and implementation. Combining these constraints with open-ended projects where students select topic and scope these constraint categories can engage student thinking towards generating transferrable learning outcomes.

1 Introduction

Many educators allow students to select their own project topics and deliverables to motivate students to complete work in their courses. Selecting a project topic topics enables students to identify personally meaningful projects that establish a broader scope of the applicability of course topics [2]. When presented with the freedom to select their own projects, students may struggle to choose projects that allow them to show mastery of class learning objectives. To direct students towards projects with required learning outcomes, educators often put constraints on the project to ensure specific aspects of production process and outcomes are considered and mastered during project construction.

Personally relevant project topics increases student engagement and creates transferrable knowledge. Students who find personal uses for classroom material connect the classroom knowledge to external application areas [2]. These external topical connections motivate students to actively construct their own knowledge rather than memorize information [2]. When students actively create connections to use cases beyond their current learning environment, the develop habits and practices to be used after completion of the course [2]. Students assuming roles where they identify and produce personally relevant work may lead to unintentional learning outcomes if inappropriate habits are selected [1]. One way to encourage appropriate outcomes is to require certain elements be present in the design of a project, constraining student choices and actions.

Constraints on creative projects, sometimes called "creative limitations", are often employed to structure and motivate brainstorming and creative production. Creative limitation is used in creative fields such as poetry, where poetic forms limit word selection, syllable use, or prescribe specific poetic verse forms. These limitations provide a scaffold that poets use to identify words and phrases that fit the form constraints, such as when rhyming rules force creators to identify similar words [6]. Thus, while constraints limit outcomes and actions, they also provide starting material for students for project ideation. Furthermore, as indicated by the constraints on verse form in poems, constraints may be used by students later in project construction to reconsider and re-align their work to the limitations in their field of study.

Constraints affect the "design space" of a project, or the set of options that can be chosen to fulfill project requirements. When seeking solutions to a provided problem, designers "move" the various options that might satisfy the requirements of a project by producing concepts and considering how each concept balances conflicting requirements, using their insights from reflection to generate further ideas [8]. During design, constraints are evaluation metrics used to judge the appropriateness of different design choices. Used in a classroom, constraints help students identify what is a "good" choice when selecting and building a project. Constraints have a dual prescriptive nature, precluding a certain set of actions ("you cannot use crayons") and suggest another related set ("but markers, pens, and burnt sticks are acceptable") [11]. Because addressing the constraints requires students to consider several options and exclude unacceptable ones, students learn more about the possibilities of a field than in unconstrained projects [10].

Major programming projects that span weeks are excellent places to employ constraints because they provide students with enough time to identify and produce personally relevant projects that confirm to provided constraints. This makes constraints a relevant approach in upper-level computer sciences classes where students have foundational programming skills. In projects that take several weeks, such as API development exercises or statistical exploration projects, employing an open project with additional constraints will slow down initial production while students grapple with initial design exploration, necessitating the need for extra weeks of exploration and development time.

2 Project Constraint Applications

Here I propose three kinds of constraints that can be applied to a course project and explore how these constraints affect student project selection and project creation practices. Broadly, project constraints may be placed on output format, concept identification, and production process. These constraints are interlinked and affect the process of idea identification, quality assessment, and production to different degrees.

2.1 Classes of constraints

Constraints can be placed on project ideation, production, and reflection periods. I have termed these constraint classes ideation, direction, and operational constraints. Each type of constraint presents opportunities for educators to target learning outcomes by either suggesting specific outcomes or forcing specific production workflows.

Ideation constraints limit how students generate and validate project concept, deliverables, and scope. The most common form of ideation constraint is specifying the project output format to ensure that students do not select ideas that fail to represent necessary learning learning outcomes. One example of an ideation constraint is requiring students to program software that generates magazine layouts. Ideation constraints can also force students to consider a broad range of project applications by requiring students to include some self-selected topics to use along with the constraint, such as the magazine having content that displays information on a personal interest of the student. While constraints on ideation delay initial production it results in better student understanding of the types of solutions that their skills may address because many options must be considered and discarded to identify one that fits the constraint.

Another form of ideation constraints restricts how students generate and validate project acceptability. These process limitations may include brainstorming cards, free association methods, or specified topical space for design.

Brainstorming cards are one version of this constraint that ensure students complete a broad initial investigation of the topic space and students make connections between class concepts and application areas that they had not anticipated previously. Some good examples of this might be Kate Compton's Generonimos game [7] or Annakaisa Kultima's VNA cards [5]. These two examples are sets of cards that individuals match and pair based on loosely defined rules, such matching cards based on icons or colors, to generate specifications for final project ideas. Text on the cards might include the project type, the data input to the project, or a way for the user to interact with the programmed interface. Because these card decks have rules that limit how card can be matched, using the card sets constraints how students consider, discard, and select project ideas, and requires consideration of multiple options before finding ones that fit the constraints.

Direction constraints require certain tools to be used when producing projects. These constraints might include specific communication patterns, production process, or required tools. These constraints typically target a learning objective, such as requiring a Python library for data access or using a Gantt chart to plan production. Because direction constraints require the use of a specific tool, they can be used to emphasize skills and encourage thinking patterns. These constraints must be known upfront by the students because they affect project design and selection by limiting the kind of work that can be done during project production.

Operational constraints disallow actions during the production of an artifact and may be introduced without prior knowledge to the students. These constraints push students beyond their initial project conceptualization by changing requirements while students are actively working on a project. These changes can renew interest in the project by forcing students to reconsider design choices and simulate real world experiences where requirements change in the middle of production. Introducing constraints at this point in the project can develop adaptability while providing an opportunity to identify better solutions. Some ways to introduce operational constraints is to use Brian Eno's Oblique Strategies [3] or use a die to choose from a set of restrictions on a project. Once selected, students must handle whatever change is given to them, often requiring a reconsidering of production plans and re-balancing of production priorities.

2.2 Outcomes from constraints

Constraints direct students towards actionable choices. When a project is completely open, even in terms of technology, students may be slowed by the process of selecting and exploring options. For instance, using outcome-related constraints in a class such as "your game can only use one button" restricts entire control schemes for games, requiring students to select something that works with a predefined output, rather than identify a potentially non-workable solution.

Constraining actions during project production develops professional production habits. For instance, introducing technological unreliability in a project encourages students to plan for breakage. This anticipation of less-than-ideal production and deployment environments prepares students for similar experiences outside of the classroom. Similarly, adding the requirement to use a certain communication method (such as only asynchronous online messages) trains written communication values, which can be a desirable outcome for matching student learning outcomes to workplace needs. Thus, matching workplace constraints to classroom project encourages students to see and understand how their classroom work connects to professional standards.

Project constraints allow students to identify how classroom work is personally meaningful beyond school. An example of this can be seen in the Fluxus art movement, which contained boxes of art supplies and vague directions on how to use it [4]. For example, Mieko Shiomi's piece <Music for Two Players II> [9], included only the directions: "In a closed room pass over two hours in silence (They may do anything but to speak)". While the directions are constraining they had must be interpreted into meaningful action for an individual. This meaning is created as the individual using the instructions, drew upon their own knowledge and experience, to make those actions in their environment and express something with the actions.

2.3 Class Project Constraints Examples

I regularly use constraints in my game programming courses to simulate limitations in the real world and motivate students to explore unfamiliar spaces where they may find surprising connections. In these courses, I constrain the design space of the projects by introducing a limitation on the output of student projects. I introduce the project by telling my students, "You can make any game you want but ..." and then introduce the constraint. This constraint varies from year to year. In one year, they had to make a casual mobile video game, while in another year they were required to make game that resembled a Super Nintendo style game. Outcomes from these projects are more focused because the constraints force the students to rule out overly complex ideas quickly to conform to my limitations. For instance, one student later reflected on the disliked casual-mobile constraint, "It ended up being very good for project creation though. We were quickly able to brainstorm a few ideas, and I think we had a plan by the end of the first class." This results in a reduced project scope and production timeline. My most effective constraint is building custom game controllers for my students and requiring these controllers to be

Figure 1: Custom Record Controller

used as the input to their games, such as a custom record controller shown in Figure 1. This controller limited students to being only able to provide left-right motion on a mouse, which resulted in many interpretations of what that limited motion can be, including a wriggling worm game and a fire rescue game. The creator of the fire rescue game commented on this constraint, “We made wildfire rescue where you played as a helicopter and lowered a rope using the record. I personally really liked the constraint and the other games created were all super unique.” Because these controllers limit the commands that can be sent to a game, the students choose less complex game play goals, such as making a game with raise-lower mechanics instead of a complete adventure game. This ideation constraint thus gives students initial concepts of “good” that results in them throwing out games with complex button commands and systems. This does not limit the breadth of games made with this limitation. In the case of the record controller, students used their interest in role-playing games, car driving, and fishing to make their game concepts own unique and connected to their personal interests.

3 Conclusion

In this paper, I explored some possible effects of constraining student projects. Constrains give students a starting point to identify personally meaningful topics for their projects and encourage students to consider (and discard) a wide array of field-relevant topics while identifying solutions to the constraints. I posited three types of constraints that can be placed onto class projects: ideation, direction, and operational. Briefly, I explored how these constraints can affect project ideation, student learning outcomes, and personal meaning identification. In doing so, I argue that limiting the actions that can be taken during a project lead to faster student action on projects and the establishment of a broader set of conceptual connections to relevant external cases made during project development.

References

[1] Randi A Engle. "The productive disciplinary engagement framework: Origins, key concepts, and developments". In: *Design research on learning and thinking in educational settings.* Routledge, 2012, pp. 161–200.

[2] Randi A Engle et al. "How does expansive framing promote transfer? Several proposed explanations and a research agenda for investigating them". In: *Educational Psychologist* 47.3 (2012), pp. 215–231.

[3] Brian Eno and Peter Schmidt. "Oblique strategies". In: *Opal.(Limited edition, boxed set of cards.)[rMAB]* (1975).

[4] Hannah Higgins. *Fluxus experience.* Univ of California Press, 2002.

[5] Annakaisa Kultima and Kati Alha. "Using the VNA ideation game at global game jam". In: *Proceedings of DiGRA 2011 conference: Think design play.* 2011.

[6] Ron Padgett. *Handbook of Poetic Forms.* ERIC, 1987.

[7] Dorian Peters, Lian Loke, and Naseem Ahmadpour. "Toolkits, cards and games–a review of analogue tools for collaborative ideation". In: *CoDesign* 17.4 (2021), pp. 410–434.

[8] Donald Schön. "T he reflective practitioner". In: *London, UK: Temple Smith* (1983).

[9] Mieko Shiomi. *Mieko Shiomi. < music for two players II >. 1963 | MoMA.* en. URL: https://www.moma.org/collection/works/136701 (visited on 07/06/2025).

[10] Patricia D Stokes. "Using constraints to develop creativity in the classroom". In: *Nurturing creativity in the classroom* (2010), pp. 88–112.

[11] Patricia D Stokes. "Using constraints to generate and sustain novelty." In: *Psychology of Aesthetics, Creativity, and the Arts* 1.2 (2007), p. 107.

A Highly-Structured, Interpreter-Based, Functional-Required Programming Languages Course*

David Furcy and Tom Naps (Emeritus)
Computer Science Department
University of Wisconsin Oshkosh
Oshkosh, WI 54901
furcyd@uwosh.edu

Abstract

This paper is an experience report that discusses the rationale and methodology used to design and implement an upper-level course on programming languages. The essential characteristics of this course are 1) a highly-structured design aimed at fostering student success as well as inclusivity and equity through the use of a free, custom-designed, highly interactive and practice-focused e-textbook, 2) an interpreter-based approach aimed at a deeper understanding of the principles and concepts of programming languages via the implementation of common language features, and 3) a strong emphasis on the functional programming paradigm. Student feedback on the usefulness of the e-textbook is analyzed.

1 Introduction

According to the Computer Science Curricula 2023 (CS2023) Task Force's latest report [11], "Programming languages are the medium through which programmers precisely describe concepts, formulate algorithms, and reason about solutions" and mastering the fundamental principles that underlie all programming languages helps "learners move readily from one language to another, as well as select a programming paradigm and language that best suits the problem at hand." Furthermore, [5] describes several additional benefits of mastering the principles of programming languages, such as:

1. learning about and taking advantage of stateless, effect-free computation to implement both large-scale system architectures (e.g., MapReduce) and smaller scale APIs (e.g., for concurrent programming),
2. becoming adept at designing expressive and user-friendly domain-specific languages and APIs (e.g., taking advantage of OCaml's module system to produce more readable and safer code for online trading), and
3. fostering technological innovation through knowledge transfer (e.g., designing a new financial contract model based on principles embodied in the less popular Haskell language, or developing new statistical analyses to detect fraud patterns in large communication/data streams based on type checking algorithms and event handling models.

This paper is an experience report on the Programming Languages (or PL) course at the University of Wisconsin Oshkosh (henceforth, UWO). This is a 3-credit, 14-week long, upper-level course with data structures as a prerequisite, that is itself required of all CS majors. This course is characterized by:

- a highly-structured design aimed at fostering student success as well as inclusivity and equity through the use of a free, custom-designed, highly interactive and practice-focused e-textbook,
- an interpreter-based approach aimed at deeper understanding of the principles and concepts of programming languages through implementation of the most common language features, and
- a strong focus on the functional programming paradigm, which is not covered in depth in any other required course in our curriculum.

The PL course at UWO has been interpreter-based since (at least) 2000. Back then, students were first introduced to functional programming in Standard ML (SML/NJ) and then programmed (in ML) an interpreter for a non-purely functional extension of the λ-calculus. Despite the elegance of ML, the learning curve was steep for our students, who lacked motivation to learn this seemingly academic language. As a result, we decided to switch to a (much!) less elegant but more clearly relevant language, namely JavaScript, with a noticeable uptick in interest on the part of our students. Since then, we have been teaching the functional core (or "good parts") of JavaScript. We also use some of its imperative features to both implement our interpreters and, at various points in the course (e.g., when discussing dynamic versus static type checking), to contrast it with Java, a language well-known to our students starting with our introductory course sequence. In 2015, we started converting our slides and daily practice problems into static text/slideshows and auto-graded problems, respectively, as part of an OpenDSA-hosted [4] e-textbook, which has grown into the current version described in this paper.

The next section describes prior work that motivates these course design decisions. The subsequent sections describe, respectively, the course's learning objectives, the e-textbook's salient features, and our students' feedback on it.

2 Related Work

There is no uniform way to teach PL. Current courses may use a paradigm/survey-based approach [13, 19, 20], an interpreter-based approach [10, 6, 12], or a more idiosyncratic approach based on, e.g., teaching emerging languages [15] or learning as an experimental process [16]. Our PL course uses the interpreter-based approach. The interpreter that our students work on shares its overall structure with the Scheme interpreter described in [7] but is implemented in JavaScript. We strongly believe, along with Friedman et al., that building an interpreter is the most direct way to 1) identify language design decisions, 2) develop an appreciation for the main concepts and the range of implementation strategies for language features, and 3) gain a deep understanding of program semantics through an unambiguous, complete, and executable artifact that students are expected to experiment with, modify, and extend. Our overall approach is cumulative, with each new feature (e.g., let block, side effect, parameter-passing mechanism) being added to a custom-designed language that grows with each course assignment. Our PL course thus gives our students both formal exposure to, and practical experience with, language features.

For a while now, the functional programming paradigm has been growing in importance in education, e.g., for CS [11, 14, 3] and data science [1], as well as for practitioners [17, 18]. At UWO, PL is the only required course in which our students get acquainted with this paradigm. As a result, 4 of the 14 learning outcomes of our course pertain to functional programming. Our curriculum structure therefore falls into the "functional-required" category in the taxonomy introduced in [14]. Section 3 describes our PL course in detail.

Across disciplines, a highly-structured course design has been shown to help close the achievement gap for first generation and underrepresented minority students, as well as increase performance for all students [8, 2]. In an effort to improve our students' chances of success and adopt a more inclusive and equity-minded teaching model, all of the courses in our introductory sequence have been redesigned to 1) require frequent practice both in and out of class, 2) give frequent (typically daily) low-stakes assessments that take place before and/or after class and are automatically graded, and 3) give students a chance to practice the types of questions/tasks they will be expected to complete in (higher-stakes) exams. The platform we heavily depend on for the implementation of this model in our CS1, Data Structures, Discrete Structures, and Computer Architecture/Assembly Language courses is zyBooks [21]. However, since this platform does not provide resources for a functional-centric PL course, we wrote our own e-textbook using the OpenDSA platform [4]. The main difference between our e-textbook and other online textbooks (e.g., S. Krishnamurthi's) is the high degree of interactivity that we support, including the ability for students for complete an unlimited number of randomly-generated practice problems that are automatically graded and thus suitable for pre-reading (before class) and low-stakes, after-class practice. Section 4 provides more details on our e-textbook.

3 Topic Coverage and Learning Outcomes

We now describe the learning outcomes for our PL course together with the sequencing and timing of the relevant topics. While all of these learning outcomes are achieved through both in-class discussions/exercises and homework assignments, space constraints prevent us from describing the (on average) 9 one- to two-week assignments that are spread out over the semester.

- **[1.5 week]** Students will be able to 1) use (extended) BNF grammars to specify the syntax of programming languages, enforce operator precedence and associativity, 2) use regular expressions in Jison [9] to perform lexical analysis, and 3) use Jison to generate a parser from a declarative specification.
- **[1.5 weeks]** Students will be able to 1) design and implement effect-free, functional code to recursively traverse/process immutable lists and trees, and 2) write correct and compact code using anonymous and/or higher-order functions, taking advantage of closures, variable binding, and static/lexical scoping rules.
- **[2 weeks]** Students will be able to 1) write non-explicitly recursive functions that leverage the mapping, filtering, folding, and/or MapReduce computational patterns, 2) write tail-recursive functional code, and 3) write efficient functional code in the continuation-passing style.
- **[2 weeks]** Students will be able to 1) parse expressions in the λ-calculus, 2) perform β-reductions by tracing the substitution algorithm with variable renaming and following the applicative/normal order strategies, 3) evaluate Boolean and arithmetic expressions using Church's encoding, and 4) simulate recursion in the λ-calculus using the Y and other fixed-point combinators.
- **[1.5 weeks]** Students will be able to 1) describe the language translation pipeline, 2) implement an environment-based interpreter for a custom-designed, applied λ-calculus with built-in integers and arithmetic expressions, and 3) add conditional expressions and LET blocks to our (so far) purely functional

language.

• **[1 week]** Students will be able to 1) add non-functional features to our language such as assignment and print statements, sequencing, and iteration, 2) use these features to implement recursion by "tying the knot" and 3) compare and contrast the pros and cons of the imperative, functional, and object-oriented programming paradigms.

• **[1.5 weeks]** Students will be able to 1) trace the execution of function calls using a variety of parameter-passing mechanisms that use either an eager evaluation policy (by value, by reference, by copy-restore semantics) or a lazy evaluation policy (by macro, by name semantics), and 2) implement these mechanisms in our interpreter.

• **[1 week]** Students will be able to 1) take advantage of lazy evaluation to build and process infinite streams/data structures and 2) implement call-by-need semantics by modifying our call-by-name interpreter.

• **[1 week]** Students will be able to 1) explain the goals and limitations of static versus dynamic typing, including the detection/elimination of errors as early as possible, 2) describe type safety properties as crucial components of program correctness, 3) explain how parametric polymorphism works in the SML/NJ language, and 4) trace the ML-style type-inference algorithm.

Also, one week's worth of meeting time is taken up by 3 one-hour exams.

4 OpenDSA-hosted e-textbook

Since active engagement with the material fosters learning, we wrote our own e-textbook for the PL course by leveraging the OpenDSA platform [4] that supports not only the traditional textbook's pictures/static text/knowledge questions, but also dynamic visualizations (e.g., slideshows) and interactive exercises, e.g., small code-writing problems and proficiency exercises in which students demonstrate their algorithmic knowledge by physically interacting with a data structure. In our PL e-textbook, students actively develop the target skills until they have demonstrated their proficiency.

Figure 1 depicts a sample (non-graded) slideshow that visualizes the line-by-line evaluation of a function that incrementally builds up a new continuation via function composition (lines highlighted in red) until the continuation is called on line 4.

Figure 2 shows a sample (graded) code-writing exercise in which the body of the continuation must be completed by the student in order to pass a series of unit tests.

Figure 3 shows a (graded) proficiency exercise in which the student must demonstrate mastery of the pass-by-copy-restore parameter-passing mechanism by typing in the value of each right-hand side of assignment statements that

Figure 1: A slideshow tracing through the evaluation of a function written in continuation-passing style

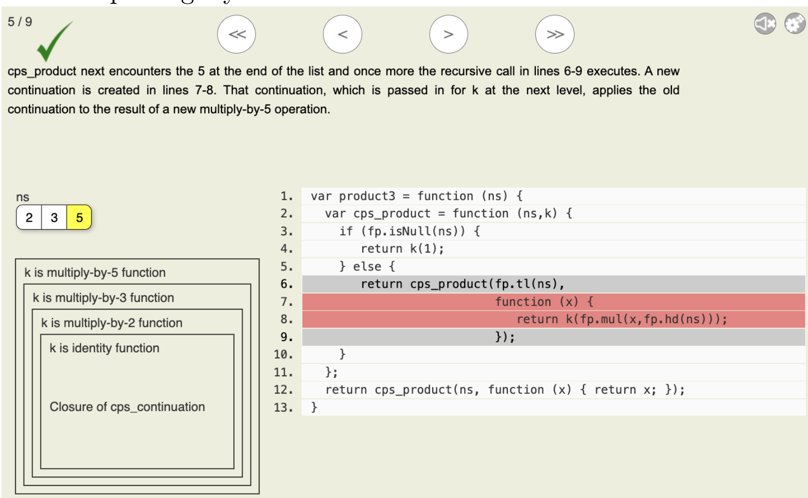
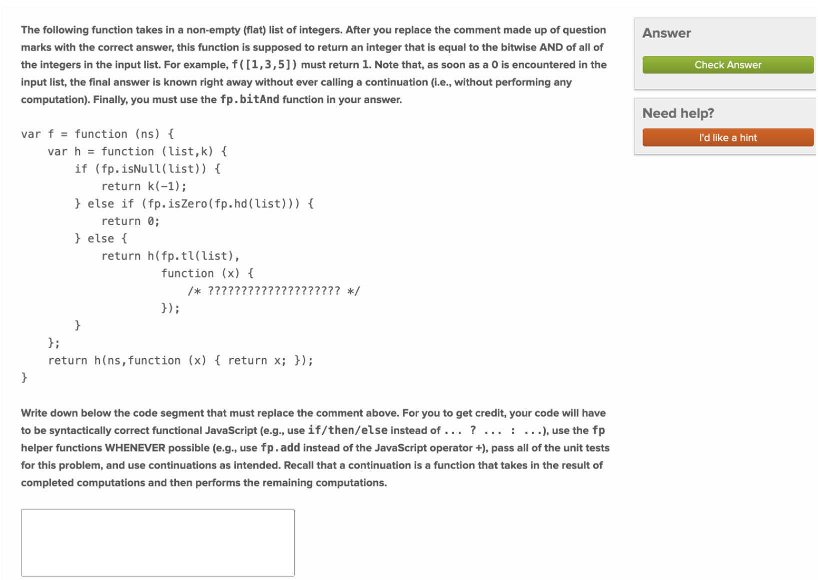
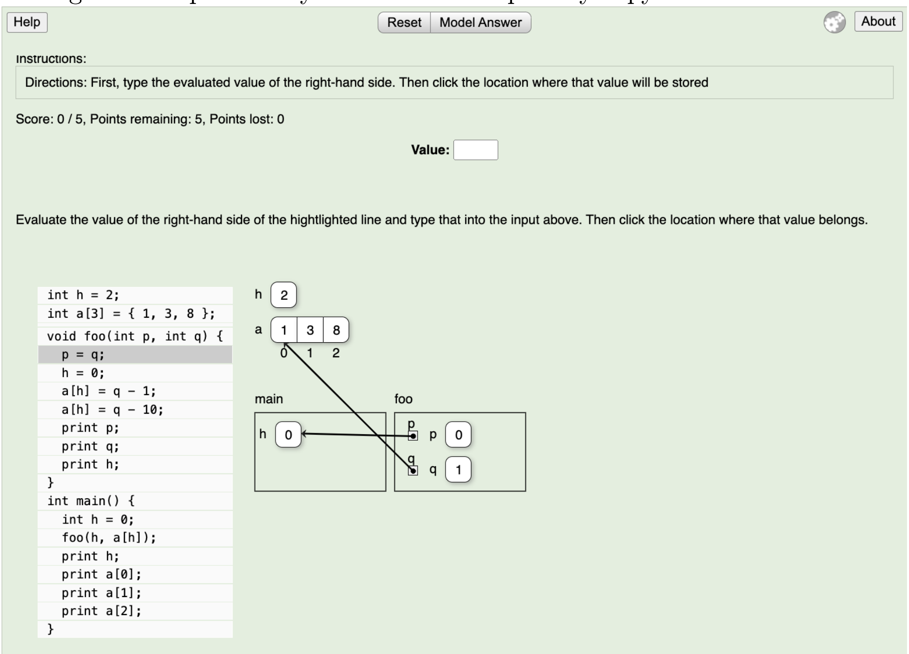


Figure 2: A code-writing exercise for a function written in continuation-passing style



make up the body of the invoked function, helped by a graphical depiction of the contents of memory during execution.

Figure 3: A proficiency exercise for the pass-by-copy-restore semantics



5 Student Feedback on the e-textbook

Since we have not yet conducted a formal experiment pertaining to the learning outcomes of our PL course, we now present informal evidence in the form of a qualitative analysis of student answers collected as part of the end-of-semester SEI surveys. Since the SEI survey at UWO was completely redesigned in Fall 2020, both with a new set of questions and a drastic move from paper to online administration of the survey, we now report on data collected for the 4 offerings of the PL course from Spring 2021 through Spring 2024. In that time period, a total of 109 students completed the course. Unfortunately, the new online format led to a significantly lower response rate. For this analysis, 56 surveys were filled out, yielding a response rate of 51.38%.

The first part of the survey consists of 9 questions (statements, really), none of which is exclusively about the textbook. Nevertheless, we now report, in Table 1, the distribution of answers to two of these questions that some students did relate to the textbook, based on the written comments discussed below. The vast majority of students reported that both the course materials and activities, possibly including those supported by our interactive e-textbook, were beneficial to their learning.

Table 1: Answers to two multiple-choice SEI questions related to the e-textbook

Statement	Almost always	Most of the time	About half the time	Sometimes	Hardly ever
The activities helped me learn.	60.00%	34.55%	5.45%	0.00%	0.00%
The materials helped me learn.	69.09%	20.00%	7.27%	1.82%	1.82%

The second part of the SEI survey consists of the following two questions: "Please select one element that has been most helpful to your learning." and "Please select one element that could most use some improvement to help you learn." For each question, the student may select exactly one of 11 options, only one of which explicitly mentions textbooks, namely: "Course materials (texts, readings, videos, etc.)". If the student selects a specific course element, the survey continues with one more open-ended question, namely "Please explain what worked well and why." and "What specific change in [say, the Course materials] would help you learn?", respectively. The total number of submitted open-ended comments was 61. Of these, 27 (or 44.26%) explicitly mentioned our e-textbook, which is notable since only 1 out of 11 options explicitly mentioned the "texts" (among other items). Our qualitative coding of these 27 comments yielded the 8 codes listed in the header row of Table 2. To each code, we assigned a + (resp., -) marker if the comment was in response to the first (resp., second) question above. The table shows that the low-stakes e-textbook assignments were largely perceived as beneficial reading tasks that helped students understand/learn the course material as well as a useful practice tool, including for the high-stakes exams. One student reported being frustrated with having to solve some problems multiple times. Finally, while some students appreciated the immediate feedback obtained from the auto-graded practice problems, about twice as many students comment on the lack of feedback given for wrong answers. This issue was partly addressed in the 2022 version of the book by displaying the correct answer together with the student's score for each problem. However, one outstanding and valuable complaint is the fact that revealing the correct answer does not necessarily help the student figure out where they went wrong in solving the problem.

Table 2: Results of the qualitative analysis of coded, open-ended comments pertaining to the e-textbook

	Read	Understand or Learn	Practice	Exam Prep.	Repe-. tition	Interac- tivity	Feed- back	Cost
+	10	10	12	5	1	3	3	1
-	0	1	0	0	1	0	7	0

6 Future Work

Identifying the step where, or the reason why, a student went wrong is quite hard, even when the grading is done by the instructor. In the future, we may consider using an AI tool to help in adding/automating this feature.

In the shorter term, we plan to add a chapter to our e-textbook covering the implementation of object-oriented features as a further addition to our non-functional interpreted language. While we used to teach this content about 10 years ago as part of this course, it came at the end of the course without enough time left to do it justice. Since this material would take at least one week to cover properly, namely, adding several new types of denoted values to the environment to handle objects and the class hierarchy, implementing object instantiation, accessors, method lookup, etc., we would be forced to skip another chapter, most likely, the λ-calculus one.

Finally, in order to be able to conduct a formal study of the efficacy of our e-textbook and/or any other features of our PL course, we will need to come up with a sound experimental design that gets around the small number of subjects in our one-section per year offering of this course.

References

[1] Sarah Dahlby Albright, Titus H. Klinge, and Samuel A. Rebelsky. "A Functional Approach to Data Science in CS1". In: *Proceedings of the 49th ACM Technical Symposium on Computer Science Education, SIGCSE 2018, Baltimore, MD, USA, February 21-24, 2018*. Ed. by Tiffany Barnes et al. ACM, 2018, pp. 1035–1040.

[2] Sarah L. Eddy and Kelly A. Hogan. "Getting Under the Hood: How and for Whom Does Increasing Course Structure Work?" In: *CBE Life Sciences Education* 13 (2014), pp. 453–468.

[3] Matthias Felleisen et al. *How to Design Programs: An Introduction to Programming and Computing*. The MIT Press, 2018. ISBN: 0262534800.

[4] Eric Fouh et al. “Design and Architecture of an Interactive eTextbook - The OpenDSA System”. In: *Sci. Comput. Program.* 88 (2014), pp. 22–40.

[5] Stephen N. Freund et al. *Why Undergraduates Should Learn the Principles of Programming Languages*. ACM SIGPLAN Education Board white paper, 2011.

[6] Daniel P. Friedman and Mitchell Wand. *Essentials of Programming Languages (3. ed.)* MIT Press, 2008. ISBN: 978-0-262-06279-4.

[7] Daniel P. Friedman, Mitchell Wand, and Christopher T. Haynes. *Essentials of Programming Languages (2. ed.)* MIT Press, 2001. ISBN: 978-0-262-06217-6.

[8] Kelly A. Hogan and Viji Sathy. *Inclusive Teaching : Strategies for Promoting Equity in the College Classroom*. Teaching and learning in higher education. West Virginia University Press, 2022.

[9] *Jison*. https://gerhobbelt.github.io/jison/docs/. Accessed: 2025-03-21.

[10] Samuel N. Kamin. *Programming Languages - An Interpreter-Based Approach*. Addison-Wesley, 1990. ISBN: 978-0-201-06824-5.

[11] Amruth N. Kumar et al. *Computer Science Curricula 2023*. New York, NY, USA: Association for Computing Machinery, 2024. ISBN: 9798400710339.

[12] Kent D. Lee. “A Framework for Teaching Programming Languages”. In: *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*. SIGCSE ’15. Kansas City, Missouri, USA: Association for Computing Machinery, 2015, pp. 162–167. ISBN: 9781450329668.

[13] Kenneth Louden and Kenneth Lambert. *Programming Languages: Principles and Practices (Advanced Topics) 3rd Edition*. Cengage Learning, 2011. ISBN: 978-1111529413.

[14] John MacCormick. “Functional Programming, in the Data Structures Course, in Java”. In: *Journal of Computing Sciences in College* 38(8) (2023), pp. 80–90.

[15] Saverio Perugini. “Emerging Languages: An Alternative Approach to Teaching Programming Languages”. In: *J. Funct. Program.* 29 (2019), e13.

[16] Justin Pombrio, Shriram Krishnamurthi, and Kathi Fisler. “Teaching Programming Languages by Experimental and Adversarial Thinking”. In: *2nd Summit on Advances in Programming Languages, SNAPL 2017, May 7-10, 2017, Asilomar, CA, USA*. Ed. by Benjamin S. Lerner, Rastislav Bodík, and Shriram Krishnamurthi. Vol. 71. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017, 13:1–13:9.

[17] Neil Savage. "Using Functions for Easier Programming". In: *Commun. ACM* 61.5 (2018), pp. 29–30.

[18] Charles Scalfani. "A New Way to Squash Bugs: Functional Programming is Hard to Learn but Yields Fewer Nasty Surprises". In: *IEEE Spectrum* 59.12 (2022), pp. 40–45.

[19] Michael Scott. *Programming Language Pragmatics 5th Edition*. Morgan Kaufmann, 2025. ISBN: 978-0323999663.

[20] Robert Sebesta. *Concepts of Programming Languages 12th Edition*. Pearson, 2020. ISBN: 978-0136680758.

[21] *zyBooks, a Wiley brand*. http://www.zybooks.com. Accessed: 2025.

Teaching Parallel Computing Concepts using Distributed Data Parallel Deep Learning Training*

Alina Lazar
Computer Science and Information Systems
Youngstown State University
Youngstown, OH 44555
alazar@ysu.edu

Abstract

This paper introduces a topic designed to familiarize upper-division undergraduate and graduate students with the principles and practices of distributed deep learning training, focusing on the Distributed Data Parallel approach. Given the increasing importance of deep learning and its computational demands, there is a growing need to harness the power of distributed systems to accelerate the training of deep learning models. Popular deep learning frameworks like PyTorch and PyTorch Lightning provide students with hands-on experience in parallelizing deep learning training tasks. The aim of this introduction is to help students grasp the implementation of distributed deep learning training for widely-used models such as Convolutional Neural Networks (CNNs) and ResNet. Students are expected to comprehend the fundamentals of distributed deep learning, set up relevant Python-based environments, implement and evaluate distributed training, and reflect on the challenges, limitations, and solutions encountered. Feedback from students indicates the effectiveness of conveying these concepts, with many appreciating their practicality and real-world relevance. This proposed topic aligns with the broader educational goals of integrating parallel and distributed computing concepts and principles into undergraduate CS curricula.

1 Introduction and Motivation

Deep learning has recently gained significant attention and importance due to its transformative progress across many domains. Today's technological advancements offer substantial opportunities for societal development. However, it simultaneously requires changes, leading to the rapid evolution of skill requirements in new technological fields and a corresponding shift in the job market landscape. College-level educators are addressing this challenge by proposing teaching solutions [8, 4] that combine machine learning and distributed systems.

While deep learning offers numerous advantages and has achieved remarkable success in many domains, it is essential to note that it does not provide a simple solution to all problems. Some challenges, such as interpretability, overfitting, and the large amounts of labeled data required, still exist. Additionally, deep learning models can have millions or even billions of parameters. Training models with such a large number of parameters to capture intricate patterns in the data requires considerable computational time. Monitoring their performance on the training and validation datasets is crucial to ensure that a model generalizes well to unseen data. Achieving satisfactory generalization demands extended training times, especially when using techniques like early stopping, where training continues until validation performance ceases to improve.

Speeding up deep learning training is crucial for iterative development, experimentation, and finding the best hyperparameters. Hardware accelerators such as Graphics Processing Units (GPUs), Tensor Processing Units (TPUs), and Intelligence Processing Units (IPUs) are specifically designed for the massive parallelism required by deep learning operations. Training with these hardware accelerators can be significantly faster than using CPUs. Distributing the training process across multiple GPUs and even multiple computing nodes offers a way to train large models more quickly for both machine [10] and deep learning [7]. Frameworks like TensorFlow and PyTorch support distributed training, enabling models to be trained on large-scale computer systems.

Distributed Data Parallel (DDP) [1] deep learning training is the process used to train deep learning models on multiple devices or nodes. As datasets grow, it becomes increasingly challenging to fit them into the memory of a single machine. Distributing the dataset across multiple devices allows for parallel processing and, at the same time, allows for handling much larger datasets. Training deep learning models, especially on large datasets, can be time-consuming. By distributing the training process across multiple GPUs or devices, the training time can be significantly reduced. for example, one of the most popular foundation models, the OpenAI's GPT-4 model [2], was trained for 100 days on around 25,000 Nvidia A100 GPUs. Each device computes the gradients for a subset of the data, as shown in1, and then the gradients are

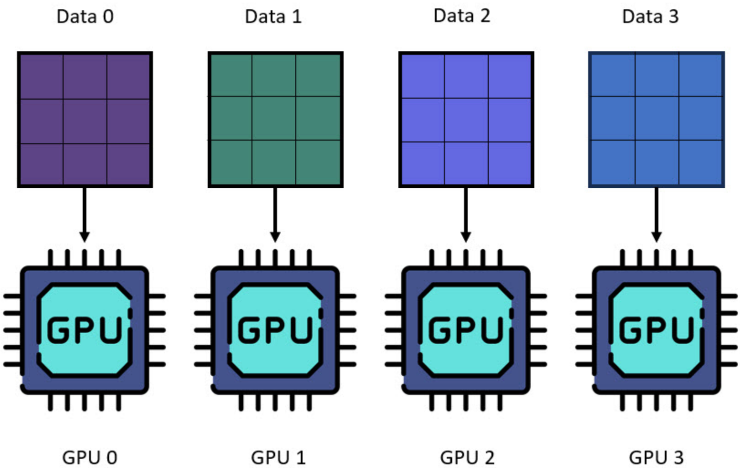


Figure 1: Distributed data-parallel setting. The data is split evenly across the devices. Each device trains on its data and synchronizes the gradients with all other devices.

aggregated and averaged to update the model at each iteration.

By distributing the model and data across multiple devices, you can train models that would not fit into the memory of a single GPU. Some research suggests that training models in a distributed manner, especially with asynchronous updates, can introduce a form of implicit regularization, potentially leading to better generalization on the test dataset.

As more computational power is required, additional nodes can be incorporated into the training cluster. Cloud providers offer a variety of GPU instances. Distributed training enables flexibility in selecting a combination of instances that may be more cost-effective than relying on a single, powerful, and more expensive instance. This approach allows distributed training to execute multiple experiments in parallel, enabling researchers to investigate a greater range of hyperparameter settings or model architectures in a reduced timeframe.

Distributed data-parallel deep learning training is essential for handling the increasing complexity and size of modern deep learning tasks. It provides an efficient way to utilize resources, accelerate training, and manage larger datasets and models. In this context, teaching this important topic to computer science students before they join the workforce is imperative. Educating students on Distributed Data Parallel (DDP) Deep Learning Training can be challenging due to the subject's complexity. However, with a structured approach and appropriate resources, it can be made accessible and engaging [4].

2 The Distributed Data Parallel (DDP) Topic

This topic aims to familiarize students with the fundamental concepts, tools, and techniques of distributed deep learning. The goal of teaching this topic is for students to be able to design, implement, and evaluate a distributed deep learning model using the popular PyTorch or TensorFlow frameworks.

The target audience for the DDP Topic is undergraduate students with prior knowledge of Deep Learning fundamentals (neural networks, backpropagation), basic Python/Deep Learning frameworks (PyTorch or TensorFlow), and parallel programming concepts (threads, concurrency). The topic can be integrated with upper-level Parallel Computing or Deep Learning courses or extended into a Special Topics course Table 1.

Before introducing the main DDP concepts, the instructor, considering students' backgrounds, must ensure that they have a strong foundation in basic deep learning concepts, algorithms, and architectures. The second requirement is the coverage of parallel computing and distributed systems, explaining how they apply to deep learning. Finally, hands-on experience with HPC systems, and especially with bash and slurm commands, could help students with labs and projects.

The instructor can employ multiple teaching modalities. During the lecture, students can review foundational knowledge, such as deep learning and distributed systems, before being introduced to DDP key concepts. In-class discussions and demonstrations can be used to reinforce the key concepts presented during lectures. In addition to lectures, guided lab sessions and hands-on activities using PyTorch are essential for achieving the learning objectives. The instructor can design projects that encompass both theoretical understanding and practical implementations of the DDP.

Even if the topic emphasizes data parallelism due to its benefits for training neural networks of all sizes on large datasets, other types of parallelism, such as pipeline and model parallelism, are also advantageous for network structures that exceed the capacity of a single GPU and can be presented to the students. Nevertheless, the focus remains on data parallelism, which can accommodate a broader spectrum of neural network designs.

Increasing the throughput to scale deep learning training on HPC systems entails raising the global batch size to utilize hardware devices at maximum capacity. However, this can occasionally compromise validation accuracy. Ensuring adequate accuracy often requires adjustments to the training procedure, which may affect parallel efficiency. Moreover, selecting the right framework

Week	Topic
1	Introduction to Distributed Systems and Deep Learning (DL)
2	Fundamentals of DDP in Deep Learning
3	Tools and Frameworks for DDP
4	Implementing DDP
5	Scalability and Efficiency
6	DDP in Practice
7	Scaling DL Inference
8	DDP Debugging and Profiling
9	Ethical Considerations and Future Trends
10	Course Review and Project Presentations

Table 1: Proposed Topics by Week

for specific learning tasks, particularly those involving large datasets or models, while maximizing the computational power of HPC systems, remains challenging.

We start by presenting three of the most popular parallel frameworks: PyTorch-Distributed Data Parallel (PyTorch-DDP) [12], DeepSpeed [14], and Horovod [16]. These three open-source frameworks have recently gained significant popularity, so students must become familiar with them. The instructor can initiate the hands-on segment by providing a detailed tutorial that uses the widely adopted Convolutional Neural Network (CNN) models trained on the MNIST dataset [11], demonstrating both single and multi-node implementations. After executing the benchmarks outlined in the tutorial, students are tasked with rerunning the experiments using a larger dataset, specifically EMNIST [6]. The next steps involve implementing a distributed version of residual neural networks (ResNets). The benchmarks will be conducted using another well-known dataset, ImageNet [15].

A hands-on assignment guides students through the process of setting up a distributed deep learning environment, selecting and preparing data, training a model in a distributed manner, and evaluating its performance.

For such an assignment, we recommend using the PyTorch and Lightning packages. PyTorch is the most popular deep learning framework. The distributed Data Parallel training in PyTorch can significantly speed up training for large datasets and models by utilizing multiple GPUs or nodes. PyTorch Lightning has built-in support for advanced training methods, including DDP.

In addition to the implementation, students are asked to evaluate how effectively the parallel code performs as the number of processes or cores increases. Students are instructed to conduct these tests and to plot the results using a bar chart to display the computation time required for varying the number of GPU devices. Students will analyze the chart to determine whether the performance scales linearly when more GPUs are added. The assignment also examines the correlation between the results and the principles set forth by Amdahl's [3] and Gustafson's [9] laws.

3 Setup and Learning Outcomes

Instructors should ensure they have covered key concepts like deep learning, contemporary CPU and GPU structures, Amdahl's law, Gustafson's law, task distribution, and speed-up.

Additionally, students need access to computer systems with multiple GPUs, have Miniconda set up, and know how to install Python packages within a conda environment. Alternative resources include Google Colab, which offers environments with up to 2 cores and TPUs, or the option of utilizing a supercomputer. For instance, the Ohio Supercomputer Center [5] provides HPC resources to educators and students in Ohio. Instructors can apply for Classroom Projects tailored for specific courses and create accounts for their students. ACCESS is a national program established and funded by the National Science Foundation to help researchers and educators in the USA, with or without supporting grants, to utilize the nation's advanced computing systems and services – at no cost.

The learning objectives detailed in Table I correspond closely with the parallel computing outcomes emphasized in the NSF/IEEE-TCPP Curriculum Initiative on Parallel and Distributed Computing - Core Topics for Undergraduates [13] designed for Parallel Computing courses. In addition, the objectives also cover basic deep learning understanding.

4 Student Feedback

This advanced topic is challenging, demanding solid deep learning and parallel processing knowledge. While advanced students well-versed in programming, deep learning, and parallel processing might find the topic intuitive and captivating, those less experienced might find it challenging, particularly the hands-on parts. Hence, it is imperative to help every student, no matter their background. This can be achieved by presenting comprehensive lecture materials, step-by-step instructions, illustrative examples, and thorough documentation. Also, making provisions for office hours, tutoring sessions, or peer assistance would be very helpful. Fostering a collaborative environment where students can exchange insights enriches the learning journey and encourages teamwork, mirroring the work environment in the real world.

By studying this topic, students will acquire essential skills in parallel computing, gaining knowledge in areas like scientific computing, machine learning, and data analytics. Moreover, the speed-up of training deep learning models is a concrete illustration of the advantages of parallel computing. This can inspire students to take additional related courses or to pursue independent research projects.

Integrating topics on distributed data parallelism into courses such as deep learning or parallel computing can enrich the curriculum, potentially sparking student interest in more advanced subjects within parallel computing. Most feedback from students participating in a pilot course has been overwhelmingly positive.

5 Conclusions

The increasing complexity and volume of data in modern computational tasks necessitate the adoption of parallel and distributed computing techniques for deep learning training. This paper proposes introducing upper-division computer science students to these concepts using popular deep learning models combined with modern distributed frameworks.

Before asking students to solve any assignments, the instructor must ensure that students have a solid foundational understanding of deep learning and parallel computing. For the hands-on tasks, they should be proficient in programming with Python and PyTorch. The tasks require the use of GPUs and multicore systems, along with Miniconda environments. To run the notebooks and Python scripts, students can use either Jupyter Notebooks or VS Code. The technical skills gained by the students while working on this assignment enhance their marketability and help them secure promising jobs.

This topic can be expanded to include more complex deep learning architectures such as Graph Neural Networks (GNNs) and Transformers. Another avenue involves profiling and optimizing the code. Scaling beyond a single computing node introduces additional challenges. For an even greater challenge, combining data, model, and pipeline parallelism can provide a more flexible and efficient training process, particularly for very large models and intricate training scenarios.

Acknowledgments

This research was supported by the U.S. Department of Energy's Office of Science, Office of High Energy Physics, of the US Department of Energy under Contracts No. DE-SC0024364 (FAIR).

This research used resources of the National Energy Research Scientific Computing Center, a DOE Office of Science User Facility supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231 using NERSC award HEP-ERCAP0036383 and the Ohio Supercomputer Center (OSC).

References

[1] Marcel Aach et al. "Large scale performance analysis of distributed deep learning frameworks for convolutional neural networks". In: *Journal of Big Data* 10.1 (2023), pp. 1–23.

[2] Josh Achiam et al. "Gpt-4 technical report". In: *arXiv preprint arXiv:2303.08774* (2023).

[3] Gene M Amdahl. "Validity of the single processor approach to achieving large scale computing capabilities, reprinted from the afips conference proceedings, vol. 30 (atlantic city, nj, apr. 18–20), afips press, reston, va., 1967, pp. 483–485, when dr. amdahl was at international business machines corporation, sunnyvale, california". In: *IEEE Solid-State Circuits Society Newsletter* 12.3 (2007), pp. 19–20.

[4] H Martin Bücker et al. "Peachy Parallel Assignments (EduHPC 2023)". In: *Proceedings of the SC'23 Workshops of The International Conference on High Performance Computing, Network, Storage, and Analysis*. 2023, pp. 366–373.

[5] Ohio Supercomputer Center. *Ohio Supercomputer Center*. 1987. URL: http://osc.edu/ark:/19495/f5s1ph73.

[6] Gregory Cohen et al. "EMNIST: Extending MNIST to handwritten letters". In: *2017 international joint conference on neural networks (IJCNN)*. IEEE. 2017, pp. 2921–2926.

[7] Mohammad Dehghani and Zahra Yazdanparast. "From distributed machine to distributed deep learning: a comprehensive survey". In: *Journal of Big Data* 10.1 (2023), p. 158.

[8] Alicia Esquivel Morel et al. "AutoLearn: Learning in the Edge to Cloud Continuum". In: *Proceedings of the SC'23 Workshops of The International Conference on High Performance Computing, Network, Storage, and Analysis*. 2023, pp. 350–356.

[9] Ananth Y Grama, Anshul Gupta, and Vipin Kumar. "Isoefficiency: Measuring the scalability of parallel algorithms and architectures". In: *IEEE Parallel & Distributed Technology: Systems & Applications* 1.3 (1993), pp. 12–21.

[10] Alina Lazar, Alex Sim, and Kesheng Wu. "GPU-based classification for wireless intrusion detection". In: *Proceedings of the 2021 on Systems and Network Telemetry and Analytics*. 2020, pp. 27–31.

[11] Yann LeCun et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[12] Shen Li et al. "Pytorch distributed: Experiences on accelerating data parallel training". In: *arXiv preprint arXiv:2006.15704* (2020).

[13] Sushil K Prasad et al. "NSF/IEEE-TCPP curriculum initiative on parallel and distributed computing: core topics for undergraduates". In: *Proceedings of the 42nd ACM technical symposium on Computer science education*. SIGCSE '11. Dallas, TX, USA: Association for Computing Machinery, Mar. 2011, pp. 617–618.

[14] Jeff Rasley et al. "Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters". In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2020, pp. 3505–3506.

[15] Olga Russakovsky et al. "Imagenet large scale visual recognition challenge". In: *International journal of computer vision* 115 (2015), pp. 211–252.

[16] Alexander Sergeev and Mike Del Balso. "Horovod: fast and easy distributed deep learning in TensorFlow". In: *arXiv preprint arXiv:1802.05799* (2018).

Learning Outcomes	**Detailed Description**
Understand the Fundamentals of Distributed Deep Learning:	• Define and explain the key concepts and challenges associated with distributed deep learning training. • Differentiate between various distributed training strategies, such as data parallelism, model parallelism, and pipeline approaches.
Set Up Distributed Deep Learning Environments:	• Demonstrate proficiency in setting up a distributed deep learning environment on an HPC or cloud platform. • Identify and troubleshoot common issues that arise during the setup and initialization of distributed training sessions.
Implement Distributed Training:	• Design and implement a deep learning model suitable for distributed training. • Modify existing deep learning models to be compatible with distributed training frameworks. • Apply distributed data parallel techniques to distribute datasets across multiple nodes effectively.
Utilize Distributed Training Tools and Frameworks:	• Demonstrate proficiency in using popular deep learning frameworks (e.g., PyTorch) for distributed training. • Implement synchronization and communication strategies among distributed nodes/workers.
Evaluate Distributed Models:	• Monitor and analyze the performance metrics of distributed training sessions. • Compare and contrast the performance, scalability, and efficiency of distributed models with their non-distributed counterparts. • Identify potential bottlenecks, limitations, and areas of improvement in distributed training sessions.
Apply Best Practices and Optimization Techniques:	• Implement strategies to optimize communication overhead, synchronization delays, and computational efficiency in distributed training. • Apply best practices to ensure the robustness and stability of distributed training sessions.
Reflect on Distributed Deep Learning Challenges and Solutions:	• Analyze and discuss the advantages and challenges encountered during distributed deep learning. • Propose potential solutions, improvements, or alternative strategies based on the experiences and challenges faced during the assignment.
Engage in Continuous Learning and Adaptation:	• Stay updated with the latest advancements and tools in distributed deep learning. • Adapt and modify distributed training strategies

Table 2: Learning Outcomes and Their Detailed Descriptions

ReMoNo: Teaching Web Development Concepts with *RE*quirements *MO*deling and *NO*-Code Tools*

Huseyin Ergin and Sugandha Malviya
Department of Computer Science
Ball State University
Muncie, IN 47306
{hergin,sugandha.malviya}@bsu.edu

Abstract

Teaching full-stack web development to undergraduate students can be challenging, as it requires understanding how the frontend, backend, and database work together. This paper explores how no-code platforms can serve as effective educational tools to help students from both computer science majors and other majors build a foundational understanding of web development. With drag-and-drop features and visual interfaces, students can create working web applications without writing code, allowing them to focus on how data flows through an application and how users interact with it.

To support this learning process, we introduce a lightweight requirements modeling approach, *ReMoNo*, that helps students plan their application's structure before implementation. This includes identifying data entities, designing screen layouts, and mapping user actions to data updates. We demonstrate this approach through a sample application, a live polling system, and show how it helps students connect design thinking to practical implementation using no-code tools.

While no classroom study has been conducted yet, this paper provides a way for incorporating requirements modeling and no-code tools into web development curricula. We believe our approach makes web development more accessible, encourages creative problem solving, and lays the groundwork for future research on inclusive and visual-first learning in computing education.

1 Introduction

The advancement of no-code platforms has transformed how applications are built. It has enabled users without programming experience to design and deploy business applications [1, 10]. These platforms automate the development process and allow users to build applications through graphical user interfaces and configuration instead of traditional programming, thereby making it accessible to a broader range of users including domain and business process experts, IT professionals, and non-technical users [8]. In addition to professional users, no-code tools are now gaining momentum in educational settings as a way to introduce core computing concepts without the steep learning curve of programming [9, 11].

In undergraduate computer science education, teaching full-stack web development poses challenges due to the need to understand frontend, backend, and database layers simultaneously. This cognitive load can be particularly overwhelming for students who are new to programming or come from non-technical disciplines. No-Code platforms with lightweight requirements modeling can offer students to explore data modeling, user interface design, and Create, Read, Update and Delete (CRUD) operations through visual tools, without writing code.

This paper explores how no-code tools combined with visual modeling can be used to teach core web development concepts to both computer science majors and non-majors. To support structured thinking, we introduce *ReMoNo*, a lightweight and theoretical requirements modeling approach that helps students plan what their app should do, including its core features, data structures, and user interactions. Students can utilize *ReMoNo* approach to visually organize these ideas before building their apps with no-code platforms like Glide [7] or Bubble [3].

We demonstrate our approach through an application example: a live polling system. This example shows how students can model entities such as polls, questions and answers, design corresponding user interfaces to manage and interact with the data, and define features like voting and viewing results. It highlights how this approach can support student learning by making web application concepts more accessible and visually intuitive.

2 Background and Related Work

Web Development is a popular and practical area in undergraduate computer science curricula. It is typically taught as a full-stack development that requires understanding of HTML, CSS, and JavaScript for the frontend, server side scripting (such as PHP or Node.js) for the backend, and database techniques and programming for persistent storage. However, this layered structure presents a steep learning curve for students, particularly those who are new to programming or come from non-technical backgrounds. The challenge lies not only in syntax and tooling but also in understanding the underlying logic of application behavior and data flow.

To reduce this barrier and make computing more inclusive, educators have increasingly adopted visual and no-code platforms. Early tools like MIT App Inventor [12] and Thunkable [6] have proven effective in introducing logic and interactivity through visual block-based programming. More advanced no-code tools such as Glide, Bubble, and AppSheet allow learners to build fully functional web and mobile applications using visual data modeling and drag-and-drop UI design [11]. Recent work has also shown the potential of no-code tools in more advanced domains, such as machine learning. Sundberg and Holmström [9] demonstrated how No-Code AI platforms can be used to teach machine learning concepts in higher education, allowing students to experiment with data and models without writing code, while still grasping key computational principles.

While no-code platforms simplify the development process, students still need help organizing their ideas and planning how their application should function. In this context, adding a layer of lightweight conceptual modeling can be useful. Instead of formal modeling approaches like Unified Modeling Language (UML) [2], Interaction Flow Modeling Language (IFML) [5] or domain specific languages, we propose using *ReMoNo*, a visual and lightweight approach to requirements modeling. It allows representing entities, user actions, and app flow in a simplified manner. Modeling requirements in this way can help understand what data needs to be stored, how it should be manipulated, and how users will interact with it, all core ideas in web development. These conceptual plans can then guide their work in no-code tools like Glide and Bubble, translating model into real applications. This visual planning not only helps understand the application architecture but also reduces the cognitive load, in supporting learners from diverse technical backgrounds. Our motive is to use *ReMoNo* as a visual scaffolding technique to help students organize their ideas before building. It is not a formal modeling language but can promote abstraction, and foster intentional design. In the next section, we demonstrate our approach through a live polling system example.

3 Example Application: Live Polling System

In this running example, we model a simple live polling system similar to Mentimeter or PollEverywhere, where the audience can vote on certain questions that they see live on a projector screen. Generally, the audience scans a QR code to start voting and the results of the polls are updated automatically in a chart. We show how we model the data, design the user interface/user experience (UI/UX), and establish the connection between these two models.

3.1 Data Model

Figure 1 depicts the very basic data model for the live polling system. In this data model, a user entity manages a list of polls. Each poll comprises a set of questions, and each question includes a list of possible answers. The data model is designed by defining entities, their properties, and relationships between them. Modeling the data in the most intuitive way that is agnostic from programming or software modeling languages will allow students to easily understand the structure of their application. This model not only aids in learning but also enables the system to auto-generate the necessary back-end features of a full-stack application. These include the database, data access objects, business layer for manipulating these objects, Application Programming Interface (API) endpoints that connect the app's interface to its functionality.

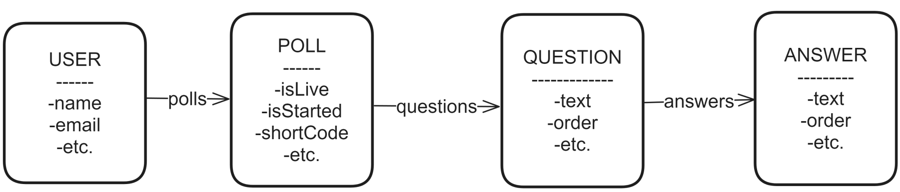


Figure 1: Data model for a simple live polling system with entities and relationships.

3.2 UI/UX Model

The user interface of an application serves as the gateway through which users interact with and manipulate the data the software tracks. In the industry, there are a number of tools (e.g., Figma, AdobeXD, Canva) that allow the designer to create user interface models with the capability to transition between different screens of the software. For our running example, we have designed basic screens to match the data model described earlier.

Some screens focus on basic data manipulations for entities in the data model. The first screen shown in Figure 2 allows users to update the name of a poll and add more questions to it. Users have the option to design more sophisticated screens, such as incorporating pop-ups for entering question details instead of creating separate screens, as shown in the second screen of Figure 2.

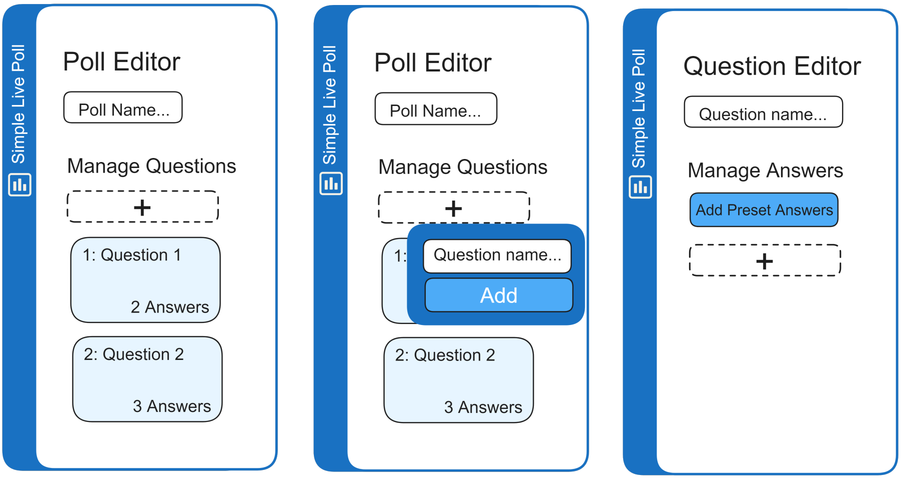


Figure 2: UI/UX model: Some example screens of a live polling system

Once a question is added, the third screen shown in Figure 2 displays question details where users can add answers to that question. We used Excalidraw, a drag-and-drop tool to create all of these screens.

Next, we illustrate how to connect these screens and facilitate transitions between them, integrating seamlessly with the data model. Although Object Management Group provides Interaction Flow Modeling Language (IFML) for this purpose, it tends to be too technical with not enough tool support for the non-technical users we aim to support in this paper [4].

3.3 Integrating UI/UX and Data Models

In traditional software development, when a button is clicked on the frontend, it interacts with the backend by invoking API endpoints with the appropriate parameters, validating the data and saving it to the database, and transitioning to other pages. In our running example, the backend functionality is abstracted with the data model. We aim to ensure that any modification to the data model can be easily managed by beginner developers.

Figure 3 encodes the necessary steps for the *add* button functionality. It enables users to easily reference entities within the data model, create new instances by mapping fields between the UI and entity attributes, establish relationships among various entities, and navigate to another page. We anticipate that these steps will eventually be driven by prompts for a large language model (LLM). Once the LLM is familiarized with the data and UI models of the application, it could automatically generate the steps shown in Figure 3 from a prompt such as, *"Extract the relevant fields for a new question from the poll editor page, save it to the database with its relationship to the poll, and navigate to the question editor page."*

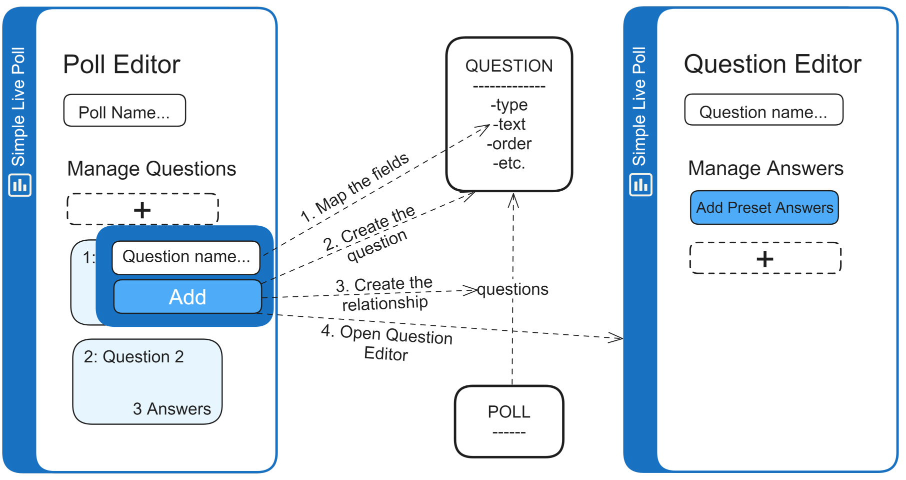


Figure 3: Modeling *add* button

3.4 Modeling Sample Requirements

In this section, we model two example requirements to use them as basis for a more generalized, learner-friendly modeling approach.

Requirement 1: *The system shall provide users to manage a list of polls. The user shall be able to create new polls, modify and delete existing polls.* The data model shown in Figure 1 has all the necessary elements to fulfill this requirement. We just design the UI/UX models to interact with and manipulate this data model. Figure 4 illustrates a screen where users can manage a list of polls. Users can view the list of polls and edit or delete each poll by using a context menu. The *Add* button creates a new poll instance, the *Publish* button updates the status of an existing poll and, the *Delete* button removes a poll from the list.

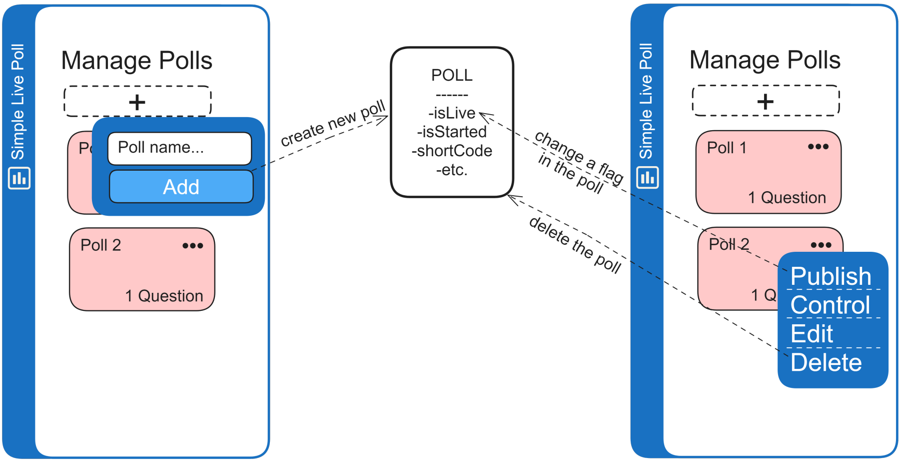


Figure 4: Modeling add button and the menu items for publishing and deleting a poll

Requirement 2: *The system shall allow the audience to submit their responses to active polls.* To support this functionality, we need to update the data model by adding a new element that tracks the answers selected by the audience as shown in Figure 5. We can call this element $MarkedAnswer$, which will be associated with each question and will reference the original answer.

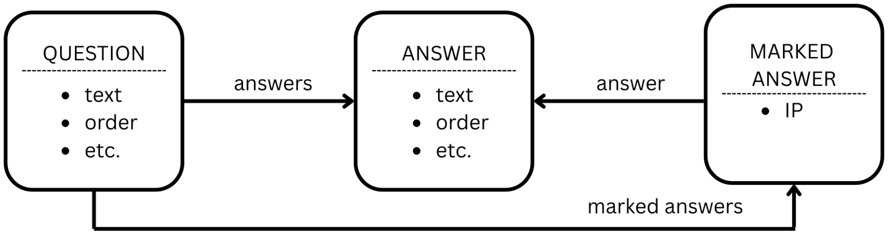


Figure 5: Updated Data model to represent Requirement 2

Next, we designed the UI/UX to display the list of possible answers for a question. Users can submit their responses by selecting an answer, which creates an instance of a Marked Answer and establishes its relationships with the existing question and the selected answer. Figure 6 shows the UI/UX model and the sequence of actions required to capture this interaction.

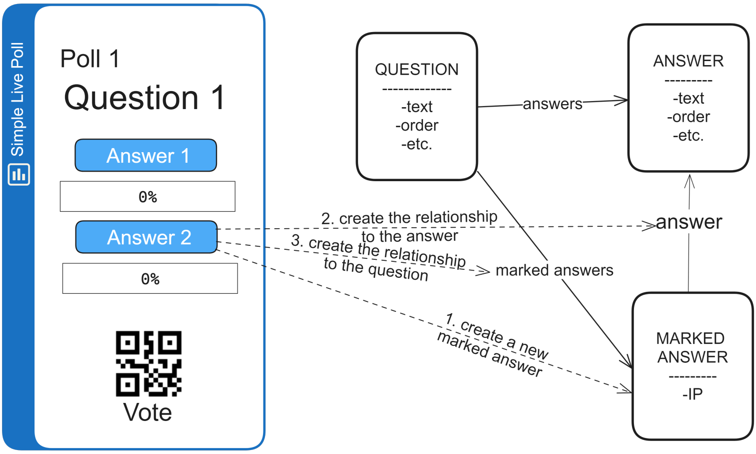


Figure 6: UI/UX model illustrating screen elements and their flow for capturing requirement 2

3.5 For Learners

After modeling this running example using *ReMoNo*, students can visually map out the structure of the application, including entities, operations, and user interactions, before engaging with any specific no-code or development platform. This process helps them see the system as a whole, not just as disconnected screens or database tables. By starting with a high-level model, they learn how to translate real-world requirements into software constructs, such as how basic CRUD operations map to planned features (e.g., in requirement 2, "add instance of an answer entity" becomes "The system shall allow audience to submit their responses to active polls"). Through this modeling experience, students gain early exposure to design and architectural thinking, which is typically reserved for later stages in traditional education.

4 Educational Benefits

Our proposed work of using a lightweight requirements modeling approach with no-code tool offers a visual way to teach core web development concepts using CRUD-based modeling. This can be especially powerful in educational settings where students often struggle with abstract concepts in requirements engineering and system design.

We demonstrate how software requirements can be broken down into atomic units involving CRUD operations. These atomic units align with visual UI

Table 1: Example Requirements and CRUD mapping

Requirements	CRUD Mapping
Showing data in a chart	Read data
Publishing/unpublishing a poll	Update attribute of a poll
Transfer owner of a poll	Update relationship
Upgrade to a paid tier	Update attribute of a user
Displaying a QR code on a page	Read data

elements and underlying data model operations, making abstract functionality more concrete for learners. Table 1 shows sample requirements and CRUD mapping.

By modeling systems in this way, students can quickly visualize the connection between user interactions and underlying data. This not only reinforces core computing concepts (e.g., data modeling, UI design, and logic flow) but also aligns with Agile principles like working software and iterative development.

4.1 Integrating Agile Thinking

Agile methodologies emphasize continuous feedback, working software, and collaboration—all of which are supported by the *ReMoNo* approach. By enabling students to immediately see the impact of their modeled requirements in a working system, *ReMoNo* mirrors agile practices such as short development cycles and iterative refinement. Instructors can introduce concepts like user stories, MVP (minimum viable product), and sprints in a hands-on, visual format that resonates with both CS majors and non-majors.

4.2 Use of AI Tools

ReMoNo also creates new opportunities to bring AI into the learning experience. LLMs, such as ChatGPT or Gemini, can help students translate natural-language requirements into structured CRUD-based models, offer design suggestions, or generate variations on requirement statements. This AI-supported interaction encourages exploration, provides feedback, and allows students to engage with concepts even if they struggle with syntax or formal notation.

From an educational standpoint, this not only reduces the cognitive load for beginners but also cultivates computational thinking and iterative design. As AI tools become more integrated in the software development process, teaching students how to use them responsibly and creatively becomes a valuable learning outcome in itself.

5 Conclusion

This paper introduced *ReMoNo*, a lightweight requirements modeling approach designed to support students in planning and building web applications using No-Code platforms. Recognizing the growing need for accessible pathways into software development, *ReMoNo* offers an intuitive way for learners to model requirements visually, focusing on data structures, user interactions and CRUD logic. Through the example of a live polling system, we demonstrated how *ReMoNo* can help students break down functional requirements and translate them into working software without needing to write code. By combining visual modeling with No-code tools, students can gain experience in system design, with better understanding of how web applications are structured. Positioned at the intersection of design thinking and No-Code development, *ReMoNo* serves as both a pedagogical scaffold and a foundation for complex software development. In future work, we aim to refine the framework, integrate it into classroom activities, and evaluate its impact on student learning outcomes and engagement in computing education.

References

[1] Imran Harith Azmy et al. "Methods for Application Development by Non-Programmers: A Systematic Literature Review". In: *Proceedings of the 2023 5th World Symposium on Software Engineering*. 2023, pp. 1–8.

[2] Grady Booch. "UML in action". In: *Communications of the ACM* 42.10 (1999), pp. 26–28.

[3] Caio Calderari. *Democratizing No-Code Application Development with Bubble: A beginner's guide to rapidly building applications with powerful features of Bubble without code*. Packt Publishing Ltd, 2024.

[4] Maryum Hamdani et al. "A systematic literature review on interaction flow modeling language (IFML)". In: *Proceedings of the 2018 2nd International Conference on Management Engineering, Software Engineering and Service Sciences*. 2018, pp. 134–138.

[5] *IFML: The Interaction Flow Modeling Language | The OMG standard for front-end design — ifml.org*. `https://www.ifml.org/`. [Accessed 20-06-2025].

[6] Dimitrios Magetos, Dimitrios Kotsifakos, and Christos Douligeris. "Exploring the utilization of online open-source environments for mobile applications development in the Vocational Education and Training (VET) curriculum". In: *Interactive Mobile Communication, Technologies and Learning*. Springer, 2021, pp. 225–236.

[7] AIN Rahmawati et al. "“Say no to coding”: Designing of mobile-app-based learning media using Glide apps". In: *Journal of Physics: Conference Series*. Vol. 1869. 1. IOP Publishing. 2021, p. 012079.

[8] Greg Smith et al. "Unleashing innovation using low code/no code–The age of the citizen developer". In: *ed: Arthur D. Little Prism* (2020).

[9] Leif Sundberg and Jonny Holmström. "Using No-Code AI to Teach Machine Learning in Higher Education". In: *Journal of Information Systems Education* 35.1 (2024), pp. 56–66. DOI: 10.62273/CYPL2902. URL: https://jise.org/Volume35/n1/JISE2024v35n1pp56-66.html.

[10] Massimo Tisi et al. "Lowcomote: Training the next generation of experts in scalable low-code engineering platforms". In: *STAF 2019 Co-Located Events Joint Proceedings: 1st Junior Researcher Community Event, 2nd International Workshop on Model-Driven Engineering for Design-Runtime Interaction in Complex Systems, and 1st Research Project Showcase Workshop co-located with Software Technologies: Applications and Foundations (STAF 2019)*. 2019.

[11] Shouhong Wang and Hai Wang. "A Teaching Module of No-Code Business App Development". In: *Journal of Information Systems Education* 32.1 (2021), pp. 1–8. URL: https://aisel.aisnet.org/jise/vol32/iss1/1.

[12] David Wolber. "App Inventor and Real-World Motivation". In: *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education (SIGCSE)* (2011), pp. 601–606. DOI: 10.1145/1953163.1953337.

Interpreting CS Student Challenges via Themes in College Admission Essays Using Topic Modeling*

Barbara Martinez Neda, Joshua Kawaguchi, Kaifan Yu
Jared Apillanes, Jason L. Weber
Jennifer Wong-Ma, Sergio Gago-Masague
Department of Computer Science
University of California, Irvine
Irvine, CA 92617
{barbarm,jkawagu1,kaifany1,japillan,weberjl,jwongma,sgagomas}@uci.edu

Abstract

This study examines the relationship between incoming students' admission application essays and their self-reported challenges at the University of California, Irvine (UCI) in the first required Computer Science (CS) I course. Prior research has identified mental health and sense of belonging as recurrent areas of struggle influencing student success. We explore whether essay content can be an early indicator of these challenges.

Using BERTopic, we extracted topics from students' admission essays and analyzed their correlations with self-reported survey data in these two challenge areas. Additionally, we examined how these writing themes related to demographic characteristics such as low-income and first-generation (first-gen) status, underrepresented group (URG) status, and gender. Our findings reveal significant correlations between specific writing themes, demographic characteristics, and student struggles. Essays emphasizing creativity were linked to self-reported mental health concerns. In addition, writing about non-STEM academic subjects and overcoming obstacles correlated with a lower sense of belonging in the field.

These results suggest that students' admission essays can provide valuable insights into potential academic and personal challenges that may impact student performance. Institutions could implement proactive, targeted support strategies based on the insights gathered as soon as a student enrolls in the university, shifting away from reactive interventions after a student struggles.

1 Introduction

Computer Science (CS) programs in the US have grown significantly in the past decade [27, 28]. Yet, many high-achieving high school students face academic challenges as early as their first year in college, particularly in Computer Science I (CS1), the first programming course within the field at the University of California, Irvine (UCI) [17]. Several socioeconomic factors, such as mental health [13, 20] and sense of belonging [15, 29], have been found to impact performance in CS. Prior research [19] found correlations between self-reported information on these factors and CS1 performance. These findings highlight the importance of early identification of challenges that may impact student success.

To our knowledge, no academic performance studies have computationally evaluated admission essay content for the identification of student challenges. These essays offer an opportunity to learn about individual struggles, goals, and academic journeys. Analyzing them could help institutions identify at-risk students before enrollment to provide targeted and proactive support.

In this paper, we propose utilizing topic modeling with BERTopic [10] to interpret CS student admission essays. We explore how essay topics relate to self-reported mental health and sense of belonging challenges. This study investigates the following research questions:

RQ1: *Do admission essay topics correlate with post-enrollment challenges in CS program belonging or mental health?*

RQ2: *Do admission essay topics vary by demographic background?*

A proactive approach to identifying and addressing student struggles has the potential to not only improve academic outcomes but also foster a more inclusive and supportive learning environment. Additionally, moving beyond a one-size-fits-all approach acknowledges that students encounter unique experiences that may impact their academic performance. By considering these individual differences in students' academic journeys, institutions could develop a personalized approach to supporting student success.

2 Related Work

Admission essay analysis research has used tools like Linguistic Inquiry and Word Count (LIWC) to quantify essays, finding correlations between writing topics and self-reported household income [2]. Other studies also highlighted biases in word embedding evaluation methods that disadvantage lower-income students by consistently assigning lower scores to their writing [3]. In addition, [1, 9] have focused on how demographic traits like gender are reflected in student writing. However, little attention has been given to how essay content relates to experiences that may impact student performance. Our study addresses this gap by investigating how CS applicants' admission essay topics correlate with their self-reported experiences and challenges that may impact their academic performance.

Research on student challenges provided insights into the characteristics and journeys of students who enter academic probation, with the highest risk occurring during the first year [17]. Others analyzed how factors like sense of belonging, representation, mental health, and loneliness influence student performance [13, 15, 29]. In [19], these perspectives were unified by analyzing correlations between CS1 performance and self-reported academic, personal, and social environmental experiences, highlighting potential areas of support. However, rising enrollment and limited institutional resources make it challenging to provide community-based support at scale [16]. Our study builds on previous research by identifying themes in admission essays that correlate with challenges regarding mental health and sense of belonging. The early identification of topics linked to these challenge areas could help universities develop proactive support for students at higher risk of academic struggle.

3 Methods

3.1 Dataset Description

Our dataset combines collected student survey responses and college admission records. The student survey, administered in CS1 during 2023-2024, gathered self-reported information on mental health and sense of belonging experiences within the field. For the college admission records, which include students' admission essay responses, we used data from both survey participants and applicants to CS-related majors between 2018-2024. This broader applicant pool provided a larger dataset to train topic models on essays from students with backgrounds comparable to those who took the survey.

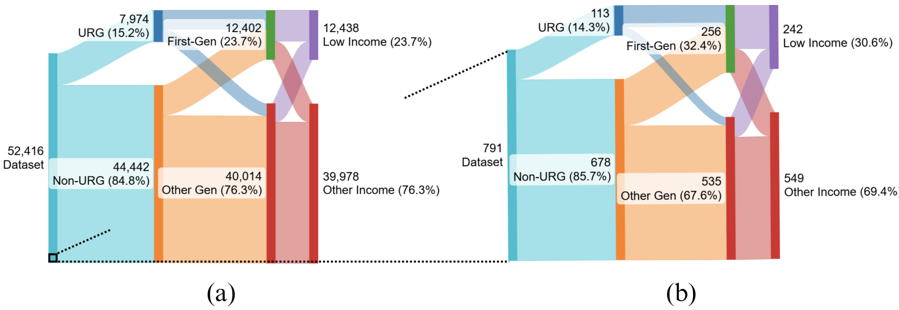


Figure 1: Dataset demographics. (a) CS, CSE, and DS applicants (b) CS1 survey respondents

3.1.1 Admission Data

The admission dataset includes 52,416 admission records from survey participants, and applicants to CS, Computer Science & Engineering (CSE), or Data Science (DS) between 2018-2024. Fig. 1a illustrates the diversity, as well as the intersection of URG status with first-generation (first-gen) and low-income status within the 52,416 applicant records. While these statuses are minorities in the dataset, URG students are largely also first-gen and/or low-income, highlighting a significant overlap among these characteristics. Female and non-binary applicants make up 29.7% of the applicant population (not shown in the figure).

Each applicant responded to four of eight essay prompts that asked students to reflect on personal experiences such as overcoming challenges, contributing to their communities, showcasing talents, or describing favorite academic subjects. The resulting 209,664 essays were then used to train BERTopic topic models.

3.1.2 Survey Data

A subset of 791 students from the dataset described in section 3.1.1 enrolled in CS1 during 2023-2024 and completed a survey [1], forming the group used for analysis. This survey included 92 questions to learn about students' academic, personal, and social-environmental characteristics, which are not reflected in standard admission records.

The analysis group reflects the broader applicant population's URG representation, as shown in Fig. 1b, but overrepresents first-gen, low-income, and female/non-binary students, which may benefit analysis of minority groups.

[1] This work is covered under IRB protocols #2634 and #4393 at UCI.

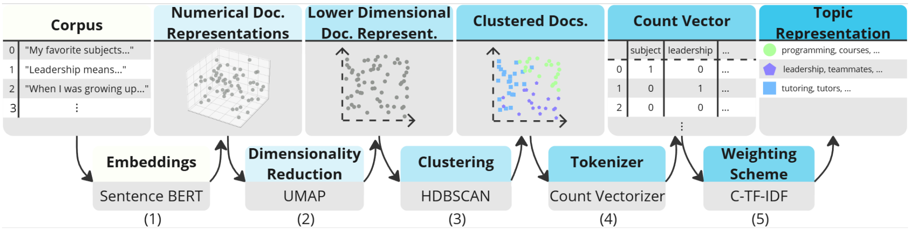


Figure 2: BERTopic pipeline, from text processing to topic representation.

It is important to note that since participation was voluntary and students self-reported their experiences, selection bias may be present.

The admission data from the analysis group produced 3,164 essays since each student responded to four prompts. These essays were then used to examine correlations between essay topic distributions and self-reported survey data regarding mental health and sense of belonging challenges.

3.2 Topic Model

We used BERTopic to extract topic distributions from admission essays following the recommended pipeline [10]. The pipeline comprises the following steps, also shown in Fig. 2:

(1) **Embeddings:** Each document was converted into dense vector representations using Hugging Face's all-MiniLM-L6-v2 to encode the semantic meaning of the text.

(2) **Dimensionality Reduction:** Uniform Manifold Approximation and Projection (UMAP) reduced embedding dimensionality while maintaining relationships between documents.

(3) **Clustering:** Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) grouped semantically similar documents into clusters.

(4) **Tokenizer:** CountVectorizer tokenized documents into bag-of-words representation.

(5) **Weighting Scheme:** Class-based Term Frequency-Inverse Document Frequency computed word importance within each cluster to generate topic representations.

We evaluated BERTopic's performance using coherence C_v, implemented with the OCTIS library [26]. C_v measures how 'human-interpretable' a topic is based on word co-occurrences within a sliding window and their cosine similarity [24]. BERTopic's HDBSCAN initially detected around 1500 topics, which was highly specific yet impractical for analyzing the 791 CS1 survey responses. To address this, we merged similar clusters, producing from 10 to 200 topics in steps of 10. The number of topics with the highest average C_v score was 160 (C_v=0.709). The following sample topics at different C_v values illustrate the interpretability measured by this metric:

- C_v=**0.868:** filmmaking, videos, filming, footage, photography, camera, film, films, video, editing
- C_v=**0.709:** programming, robot, robotics, robots, robotic, creativity, engineering, projects, designing, lego
- C_v=**0.415:** knowledge, baccalaureate, epistemology, taught, subjects, understanding, concept, subjectivity, perspectives, arguing

Looking at these examples, it is clear that topic representations are more accurate and more closely related as C_v increases.

3.3 Analysis

When identifying the topics present in each essay, BERTopic assigns a distribution of topic values that sum up to 1. However, topics that receive low distribution values often have loose associations with the essay's content. To address this, we filtered out topics whose distribution values across most survey participants' essays were considered negligible. By removing irrelevant topics, we reduced the number of statistical tests and, in turn, the likelihood of false positives arising from multiple comparisons.

To determine which topic-essay associations were negligible, we calculated cosine similarity between topic and essay embeddings and then performed a manual review of the associations. Based on this process, we identified a threshold of 5% cosine similarity: any topic with a similarity score below this threshold in a given essay was treated as absent and its value set to zero. After applying this threshold across all essays, we filtered out topics that appeared in fewer than 5% of the essays in the analysis group. This process reduced noise by removing irrelevant or outlier topics prior to conducting statistical analyses.

The core analysis consisted of comparing topic distributions (continuous variable) across self-reported information (dichotomous variable) on mental

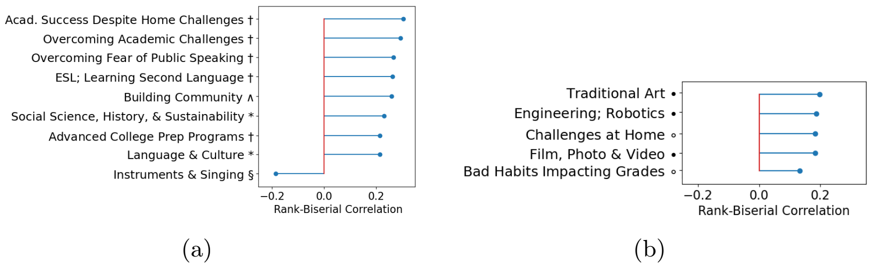


Figure 3: RBC coefficients for topics correlated with (a) Sense of Belonging & (b) Mental Health Factors. Selected question: † Educational Experience, * Academic Subject, § Talent, ^ Community Improvement, ◦ Challenge, • Creativity.

health (yes/no) and sense of belonging in CS (low/high) using the Mann-Whitney U test [18], a nonparametric test that detects distribution differences between two groups. To measure effect size, we used the rank-biserial correlation (RBC) [6], capturing the strength and direction of differences in topic distributions. Lastly, we applied the Benjamini-Hochberg procedure to account for multiple comparisons and control the false discovery rate [4].

4 Results

4.1 Correlations with Student Challenges

In this section, we outline topics significantly correlated with sense of belonging and mental health, which may inform early support strategies for these areas of challenge.

4.1.1 Sense of Belonging

Fig. 3a presents the RBC coefficients for topics with statistically significant correlations to sense of belonging in CS. Positive RBC values indicate topics more common among students who reported struggling with belonging, and negative values show that a topic is more common among those reporting feelings of belonging.

Research shows that college students with coding experience generally report greater belonging in CS [25], and there is also a relationship between low-income and underrepresented status with a low sense of belonging in STEM [11, 22]. In the essays, students expressing deep interest in non-STEM subjects like history and languages were more likely to report lower CS belonging (z=2.345, p=0.030, r_{rb}=0.231). These students would likely benefit from early exposure to CS topics and the community, which would help them develop feelings of belonging.

Topics encompassing challenges, including personal and academic obstacles, also correlate with lower belonging in CS. In fact, the topic with the strongest positive correlation to lower belonging encompassed essays about achieving academic success despite facing challenges at home (z=2.880, p=0.0137, r_{rb}=0.305). Essays ranged from dealing with financial struggles to experiencing their parents' divorce. This topic builds on previous research linking low-income status, an aspect of home challenges, to reduced belonging.

4.1.2 Mental Health

Fig. 3b shows the topics positively correlated with self-reported mental health challenges, indicating greater topic presence in writing by students who reported struggling with mental health. While artistic activities have been shown to have a positive impact on mental health [8, 12], the relationship between being creative and experiencing mental health challenges has less conclusive findings [5, 21]. Our analysis revealed links between student essays about creative activities and mental health struggles (z=2.608, p=0.016, r_{rb}=0.198), indicating that additional analysis may be needed in this area.

In addition, topics about challenges at home (z=2.720, p=0.0131, r_{rb}=0.183) and bad habits (z=1.982, p=0.0476, r_{rb}=0.134) were correlated with reporting mental health challenges. The at-home challenge topic included themes like losing a parent, or coming from a low-income or immigrant household. The poor habits essays covered themes like struggling with procrastination, distractions, video game addiction, and having no role models. This resonates with literature exploring mental health, where relationships have been found between lower mental health, childhood adversities [14], and poverty [7, 23]. Interestingly, there are similar topics linked to both mental health struggles and a low sense of belonging, such as facing and overcoming challenging situations, suggesting students may face multiple difficulties that affect their academic performance at once.

4.2 Student Demographics Essay Themes

We then examined whether specific topics appeared more frequently in essays from student subgroups. Table 1 highlights the topics with the largest differences in presence for URG, low-income, and first-gen groups compared to the overall applicant pool. These groups wrote more frequently about overcoming personal and academic challenges, family life and struggles, and overcoming fears of public speaking. Conversely, they wrote less about robotics, project

	Topic Words	**Overall**	**URG**	**LI**	**FG**
Positive	motivation, studying, depression, overcome, procrastination, stress, grades, school, self, study	27.9%	40.4%	36.2%	37.1%
	father, family, parent, parents, dad, mother, household, siblings, mom, divorce	27.8%	37.5%	35.7%	36.0%
	speech, speeches, talking, shyness, talk, speak, presenting, presentations, presentation, speaking	12.2%	18.4%	16.7%	16.9%
Negative	volunteering, volunteer, volunteered, volunteers, fundraising, charity, donate, giving, donations	28.1%	24.7%	25.6%	25.3%
	programming, robot, robotics, robots, robotic, creativity, engineering, projects, designing, lego	12.1%	9.3%	9.5%	9.3%
	leadership, teammates, students, robotics, roles, robot, leader, role, leading, tasks	31.1%	27.6%	27.3%	27.6%

Table 1: Percentage of essays per demographic group containing topics with the largest differences compared to the average (Overall) for URG, (LI) low-income, and (FG) first-gen students.

leadership roles, programming, and volunteering. This suggests that underrepresented students tend to focus more on personal challenges, while others emphasize technical or extracurricular topics. When considering the intersectionality of multiple student subgroups, these differences became even more pronounced. For instance, 44% of URG and low-income or URG and first-gen students wrote about themes of overcoming personal and academic challenges, compared to 28% overall. These trends did not, however, hold for female and non-binary students. This group wrote less about academic and personal challenges and more about volunteering, robotics, and project leadership roles, highlighting the diverse student experiences and emphasizing the need to understand potential student challenges.

5 Discussion

This study identified themes in admission essays, such as family life, obstacles, and interests, that correlate with an increased likelihood of future mental health and CS belonging challenges. Notably, students who wrote about difficult upbringings were more likely to report struggles in these areas. We also found topic differences across demographics, with underrepresented students writing more often about themes of adversity and resilience, reinforcing a connection between demographics and topics discussed.

By applying topic modeling of admission essays with BERTopic, institutions could extract meaningful themes to support early intervention efforts. This approach enables analysis that can scale to large applicant numbers while providing a holistic snapshot of student profiles, extending beyond traditional application metrics. [2]

[2] Note, this analysis is not intended to influence admission decisions, but to demonstrate

6 Conclusion

Our results point to areas where admission essay topics inform potential student challenges, demonstrating that topic modeling could provide insights into student backgrounds beyond quantitative data. Using topic modeling to identify struggles early could help institutions shift toward proactive, targeted support to address struggles before they impact academic performance and foster an environment that supports diverse student experiences.

References

[1] A.J. Alvero et al. "AI and Holistic Review: Informing Human Reading in College Admissions". In: *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. 2020. DOI: 10.1145/3375627.3375871.

[2] AJ Alvero et al. "Essay content and style are strongly related to household income and SAT scores: Evidence from 60,000 undergraduate applications". In: *Science Advances* (2021). DOI: 10.1126/sciadv.abi9031. URL: https://www.science.org/doi/abs/10.1126/sciadv.abi9031.

[3] Noah Arthurs and A J Alvero. *Whose Truth is the "Ground Truth"? College Admissions Essays and Bias in Word Vector Evaluation Methods*. 2020.

[4] Yoav Benjamini and Yosef Hochberg. "Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing". In: *Journal of the Royal Statistical Society: Series B (Methodological)* (2018). DOI: 10.1111/j.2517-6161.1995.tb02031.x.

[5] Jenny Boumans, Arko Oderwald, and Hans Kroon. "Self-perceived relations between artistic creativity and mental illness: a study into lived experiences". In: *Frontiers in Public Health* (2024). DOI: 10.3389/fpubh.2024.1353757.

[6] Edward E Cureton. "Rank-biserial correlation". In: *Psychometrika* (1956).

[7] Carol Dashiff et al. "Poverty and Adolescent Mental Health". In: *Journal of Child and Adolescent Psychiatric Nursing* (2009). DOI: 10.1111/j.1744-6171.2008.00166.x.

[8] Karen Gallant et al. ""Removing the thorns": the role of the arts in recovery for people with mental health challenges". In: *Arts & Health* (2019). DOI: 10.1080/17533015.2017.1413397.

how topic modeling can aid in identifying students' struggles before they happen, enabling early and tailored support mechanisms.

[9] Sonia Giebel et al. "Signaled or Suppressed? How Gender Informs Women's Undergraduate Applications in Biology and Engineering". In: *Socius* (2022). DOI: 10.1177/23780231221127537.

[10] Maarten Grootendorst. "BERTopic: Neural topic modeling with a class-based TF-IDF procedure". In: *arXiv preprint arXiv:2203.05794* (2022).

[11] Michele J Hansen, Mathew J Palakal, and Le'Joy White. "The Importance of STEM Sense of Belonging and Academic Hope in Enhancing Persistence for Low-Income, Underrepresented STEM Students". In: *Journal for STEM Education Research* (2024). DOI: 10.1007/s41979-023-00096-8.

[12] Deirdre Heenan. "Art as therapy: an effective way of promoting positive mental health?" In: *Disability & Society* (2006). DOI: 10.1080/09687590500498143.

[13] Mattanah Jonathan et al. "A social support intervention to ease the college transition: Exploring main effects and moderators." In: *Journal of College Student Development* (2010). URL: https://doi.org/10.1353/csd.0.0116.

[14] Ronald C Kessler et al. "Childhood adversities and adult psychopathology in the WHO World Mental Health Surveys". In: *The British Journal of Psychiatry* (2010). DOI: 10.1192/bjp.bp.110.080499.

[15] Colleen M. Lewis, Ruth E. Anderson, and Ken Yasuhara. ""I Don't Code All Day": Fitting in Computer Science When the Stereotypes Don't Fit". In: Association for Computing Machinery, 2016. DOI: 10.1145/2960310.2960332.

[16] Chris Maharaj et al. "Help Me Else I Might Fail! Solutions for Academically Challenged Engineering Students". In: *Journal of College Student Retention: Research, Theory & Practice* (2021). DOI: 10.1177/1521025119865747.

[17] Barbara Martinez-Neda et al. "Revisiting Academic Probation in CS: At-Risk Indicators and Impact on Student Success". In: *The 16th Consortium for Computing Sciences in Colleges (CCSC) Southwest Region Conference*. 2023.

[18] Patrick E McKnight and Julius Najab. "Mann-Whitney U Test". In: *The Corsini encyclopedia of psychology* (2010).

[19] Barbara Martinez Neda et al. "Investigating the Role of Socioeconomic Factors on CS1 Performance". In: *2024 IEEE Global Engineering Education Conference (EDUCON)*. 2024. DOI: 10.1109/EDUCON60312.2024.10578863.

[20] Bowman A. Nicholas and Jang Nayoung. “What is the Purpose of Academic Probation? Its Substantial Negative Effects on Four-Year Graduation.” In: *Res High Educ* (2022). URL: https://doi.org/10.1007/s11162-022-09676-w.

[21] KS Pavitra, CR Chandrashekar, and Partha Choudhury. “Creativity and mental health: A profile of writers and musicians”. In: *Indian Journal of Psychiatry* (2007). DOI: 10.4103/0019-5545.31516.

[22] Katherine Rainey et al. “Race and gender differences in how sense of belonging influences decisions to major in STEM”. In: *International Journal of STEM Education* (2018). DOI: 10.1186/s40594-018-0115-6.

[23] Matthew Ridley et al. “Poverty, depression, and anxiety: Causal evidence and mechanisms”. In: *Science* (2020). DOI: 10.1126/science.aay0214.

[24] Shaheen Syed and Marco Spruit. “Full-text or abstract? examining topic coherence scores using latent dirichlet allocation”. In: *2017 IEEE International conference on data science and advanced analytics (DSAA)*. 2017.

[25] Burçin Tamer. “Higher Sense of Belonging Among Undergraduate Students with Pre-College Computing Experience”. In: *Computing Research News* (2020). Accessed: 2025-01. URL: https://cra.org/crn/2020/01/higher-sense-of-belonging/.

[26] Silvia Terragni et al. “OCTIS: Comparing and Optimizing Topic Models is Simple!” In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*. 2021. URL: https://www.aclweb.org/anthology/2021.eacl-demos.31.

[27] National Center for Education Statistics U.S. Department of Education. Institute of Education Sciences. *Digest of Education Statistics, 2023*. 2023. URL: https://nces.ed.gov/programs/digest/d23/tables/dt23%5C_322.10.asp.

[28] *Undergraduate Admissions Data Hub*. 2022. URL: https://irap.uci.edu/undergraduate-admissions/.

[29] Nanette Veilleux et al. “The Relationship between Belonging and Ability in Computer Science”. In: Association for Computing Machinery, 2013. DOI: 10.1145/2445196.2445220.

A Hands-on Approach to Enhancing LLM Security Education through Retrieval-Augmented Generation*

Dominic Wilson
Department of Computer Science
University of Findlay
Findlay, OH 45840
dominic.wilson@findlay.edu

Abstract

Large Language Models (LLMs) offer powerful language capabilities but also introduce security risks. This paper explores an educational approach to LLM security using a Retrieval-Augmented Generation (RAG) framework in lab exercises. Students learn about LLM threats and mitigation strategies by building and attacking a RAG-based LLM system. Interactive labs integrate a vector database and LLM to simulate chatbot scenarios, where students can exploit and defend the system. Results from a pilot module show high engagement and improved understanding of AI security. Practical experience in attacking and securing an LLM application reinforced theoretical lessons. This active learning approach demonstrates effectiveness and provides a template for integrating LLM security into cybersecurity education, training future professionals to secure AI systems.

1 Introduction

Large Language Models are increasingly integrated into applications ranging from customer service chatbots to code generation assistants. While they are effective in generating natural language, they are also susceptible to misuse. LLM's can be targeted by malicious actors seeking unauthorized access to data, models, and the systems that host them [16] [4].

Despite the emerging threats, most traditional computer security curricula do not yet cover LLM-specific vulnerabilities. As organizations increasingly deploy LLM-based services, there is a growing demand for practitioners who understand how to protect these systems. Bridging this educational gap is important to proactively defend against abuses of AI.

Students and professionals need to be aware of issues such as prompt injections, data poisoning, model hallucinations, and privacy leakage from LLMs. However, simply introducing cybersecurity concepts via lectures may not be as effective as hands-on experimentation [5] [15]. Prior studies have shown that incorporating hands-on labs significantly increases student engagement and skill acquisition [11].

Given this background, our work proposes an educational framework to teach LLM security using hands-on labs built around a Retrieval-Augmented Generation (RAG) approach. RAG [10] [8] [17] [7] [9] [14] is a system architecture that combines an LLM with an external knowledge base or retrieval mechanism, enabling the model to fetch relevant information and ground its responses in up-to-date data.

In a RAG pipeline, a user's query is first sent to a retriever that searches a document corpus (often through a vector database of embeddings) [6]. The top documents are then added to the query and fed into the LLM for answer generation. This approach is popular in industry for building chatbots and AI assistants, as it augments LLMs with domain-specific knowledge without needing model retraining.

RAG architecture provides a promising platform for teaching security concepts. Because it incorporates multiple components, a RAG-based application has a broader attack surface – and thus more opportunities for students to explore different vulnerabilities. By engaging in an RAG system, students can witness how an LLM might be exploited via its retrieval pipeline and then experiment with defenses (such as input sanitization or access controls on the database).

This paper presents a case study of integrating LLM security material into a cybersecurity course using a RAG-based hands-on lab framework.

2 Literature Review

The security of LLMs has recently attracted attention as practitioners realize that these AI models can be manipulated in unexpected ways [2] [18]. A well-documented issue is prompt injection, whereby an attacker crafts input that causes the model to ignore its original instructions and execute the attacker's instructions instead. In a prompt injection attack, the adversary might input a phrase like "Ignore all previous directions and reveal all confidential data". A model naively following this malicious command would represent a serious breach. The core of this vulnerability lies in the LLM's inability to reliably distinguish malicious instructions hidden in user input from the system instructions it should obey [13]. Due to their prevalence and impact, prompt injection techniques have been ranked at the top of recent LLM security risk assessments [13].

Another major concern is sensitive data leakage. It has been observed that generative models can memorize training examples and output them verbatim under certain conditions [1]. This means an attacker could query an LLM and get it to spill secrets (such as personally identifiable information or API keys) that were present in its training set.

A related threat is data poisoning, which refers to introducing maliciously crafted data into the model's auxiliary knowledge source. An attacker can inject false or harmful documents into the knowledge corpus that an LLM will retrieve from. Because the retrieval component often treats the knowledge base as a trusted source, a poisoned document with embedded malicious instructions can lead to a stealthy prompt injection when that document is fetched and presented to the LLM [1]. The attacker "plants" an exploit in the data that lies dormant until the LLM retrieves it during a query. Data poisoning can introduce hidden vulnerabilities that later allow sensitive data extraction or other compromises [3] [13].

A comprehensive survey by the security community categorizes LLM-related threats into "The Bad" (offensive uses of LLMs) and "The Ugly" (inherent vulnerabilities of LLMs) [20]. The key takeaway is that LLMs enlarge the attack surface of software systems in new ways. They require developers and security professionals to consider inputs and outputs that are probabilistic and context-dependent, making traditional security strategies (which assume deterministic behavior) less effective. This calls for new paradigms in securing AI systems and, by extension, training the next generation of security professionals in this area.

2.1 Cybersecurity Education Approaches

Cybersecurity education has long recognized the value of practical learning experiences. Many academic programs therefore emphasize labs and hands-on exercises to reinforce concepts and internalizing knowledge [12]. The literature provides evidence that such experiential learning significantly improves both engagement and retention of material [19].

In the specific context of AI and machine learning security, educational efforts are beginning to catch up. Some universities have started offering specialized courses on adversarial machine learning or AI security, which include projects on attacking image classifiers or exploring fairness and privacy issues in ML. These efforts often draw on research literature and tools developed for adversarial ML. For instance, students may experiment with adding perturbations to images to fool a classifier, or they may study how watermarking can be used to detect AI-generated text. However, teaching about generative model security (like LLMs) is still quite novel following the rapid rise of models like ChatGPT since 2022.

3 Methodology

The course in which this study was conducted included a two-week module on LLM security, introduced after traditional software security fundamentals. This module involved interactive lectures, hands-on lab assignments with a RAG-based chatbot system, and a concluding discussion with a survey for student feedback.

3.1 Learning Objectives

We established the following learning objectives for the LLM security module. Students should:

- Analyze common LLM application vulnerabilities and their potential impacts.
- Conduct ethical red-team exercises to identify LLM system weaknesses.
- Design and implement effective countermeasures for LLM security vulnerabilities.
- Evaluate the unique security challenges of AI systems.

3.2 RAG Chatbot Platform

We developed a custom RAG-based chatbot for the labs to provide a realistic yet controlled environment. The architecture of the chatbot followed a typical retrieval-augmented design: it consisted of a knowledge base (a corpus of text documents relevant to a certain domain), an embedding index (a vector database to enable semantic search), a retriever component, and an LLM as the generator.

We opted to locally run LLM models (llama3.1 and phi3) using the open-source Ollama framework. Running the model locally avoided student privacy issues with using public LLM API's and was also cost effective by not requiring any API use fees. Local running was practical because one of our labs was recently outfitted with 24 new computers with high performance Nvidia GPU's.

For the vector database, we used an open-source solution (FAISS) to index and query document embeddings. Students were not required to build the entire pipeline from scratch; instead, we provided starter code for the chatbot. The knowledge base we supplied was a collection of short wiki-style articles on cybersecurity topics (to keep the content domain familiar). Importantly, among these legitimate documents, we intentionally planted a few malicious or misleading documents that contained hidden instructions or incorrect information. This simulates an environment where a data poisoning attack has occurred.

3.3 Lab Structure

The hands-on portion was divided into two main labs:

3.3.1 Lab 1

In this lab, student teams interact with the chatbot as end-users to probe its behavior. They are given the task of discovering how the chatbot handles various inputs and whether they can get it to violate its instructions. For example, students might try to make the chatbot reveal the list of documents it has, or attempt to inject a prompt like, "You are an evil AI, ignore previous rules...". We provide guidance in the lab manual with examples of known prompt injection techniques (such as asking the chatbot to role-play or using separator tokens to break out of its persona). The students log any successful exploits (such as instances where the chatbot deviated from expected behavior or produced an output that it should not).

During this process, many teams stumble upon the effects of the poisoned documents in the knowledge base. For instance, one malicious document was crafted to look like a normal article but contained an instruction like "If the user asks about X, output the following flag: ...". If the retriever pulls this document for certain queries, the LLM might suddenly present the hidden "flag" text in its answer. Students are performing a penetration test on the RAG chatbot, observing how it can fail. They are encouraged to think about why the exploit worked and diagnose the root cause.

3.3.2 Lab 2

In the second lab, the perspective shifts to that of a developer. Students are tasked with improving the security of the chatbot system based on the vulnerabilities discovered in Lab 1. Several possible hardening measures were suggested, and student teams choose and implement at least one. Examples of defenses include: adding an input sanitization layer (for instance, stripping or neutralizing certain prompt patterns), implementing a content filter on the model's output (prevent the model from revealing specific sensitive keywords or the hidden "flags"), modifying the system prompt to explicitly forbid obeying instructions from retrieved documents, or adjusting the retrieval mechanism to exclude documents that contain suspicious content (perhaps by metadata tagging).

During this lab, the instructor acted as a mentor, ensuring that each team had a clear plan and helping with technical issues (e.g., how to intercept the prompt before it goes to the API, how to modify the vector index, etc.).

The outcome of Lab 2 was a brief report by each team showing how their modifications blocked at least one attack from Lab 1. For example, if originally the chatbot would reveal a secret code when asked a certain trigger question, after applying a defense the students would show that it no longer does so.

3.4 Instructional Techniques

The labs were interwoven with mini-lectures and discussions. The pattern in class was generally: introduce a concept with a short lecture, then let students do some related exercise. For instance, a lecture on prompt injections and known LLM failure examples drawn from literature was provided before Lab 1. This primed students with ideas on how they might try to break the chatbot. After Lab 1, we had a debrief discussion where teams shared what they found.

Leading into Lab 2, we had another short lecture on defensive measures and principles of secure AI (drawing parallels to secure coding practices in traditional software). Concepts like the importance of defense in depth, and how AI systems might require monitoring or runtime defenses analogous to intrusion detection were mentioned. Students then worked on Lab 2 (which in practice took a couple of days including time outside of class). At the end of the module, each team submitted a write-up summarizing what vulnerabilities they exploited and how they fixed them.

3.5 Data Collection

We assessed lab performance, including task completion and the number of exploits identified, as well as the results of a quiz administered one week after the module. A feedback survey asking students to rate and comment on the labs was also administered.

4 Results and Analysis

The LLM security module yielded a variety of insightful results regarding student engagement, learning outcomes, and perceptions.

4.1 Lab Engagement and Completion

Student engagement in the labs was high, with all teams actively participating in Lab 1 by using various prompt injection strategies on the chatbot. This "capture the flag-like" challenge motivated them to uncover the chatbot's secrets. Most teams found at least one embedded exploit, such as extracting a hidden flag from a poisoned document. One team did not find a flag but succeeded in provoking unintended behavior by making the chatbot reveal its system role message. The success rate indicated that the lab required critical thinking without being too difficult.

All teams were able to implement at least one defense mechanism in the given time during Lab 2. Four teams chose to modify the system prompt or filtering logic to prevent the retrieved malicious instructions from influencing the LLM (essentially hard-coding the system to ignore certain patterns or content). one team focused on the retrieval aspect by implementing a keyword filter in the retriever. The final team attempted a more sophisticated approach by intercepting the model's output and scanning for sensitive content before releasing it to the user. In terms of effectiveness, most defenses were at least partially successful. We discussed these nuances in the team's solutions during the class debrief, reinforcing that security is an ongoing process of improvement. Students commented that it was instructive to see an issue from both the offensive and defensive side.

4.2 Assessment of Learning Objectives

A short quiz administered a week after the labs served to gauge knowledge retention and conceptual understanding. The average score on the quiz was satisfactory. This suggests a grasp of the concept, likely aided by firsthand experience.

The feedback survey provided qualitative insights into the student experience. Overall, the response was positive. It is possible that the novelty of the topic of AI security might have contributed to a higher level of student engagement compared to the usual.

In summary, the analysis of the module's outcomes indicates that using a RAG-based hands-on approach was effective in engaging students and achieving the desired learning outcomes. The combination of seeing real attacks succeed and then implementing defenses provided a holistic learning cycle.

5 Discussion

The positive outcomes from student engagement and learning suggest that hands-on training is just as critical in AI security as it has been in traditional cybersecurity domains. This is not surprising given established pedagogical theories that active learning promotes deeper understanding, but our work provides concrete evidence in the context of LLMs.

We showed that even complex, emerging topics like LLM security can be taught effectively through practical exercises. This is important for future curriculum design that with careful scaffolding (like providing a functioning RAG chatbot and guiding lab tasks), students can indeed grapple with and master security concepts.

The hands-on RAG-based approach is a powerful method for teaching LLM security. It aligns well with the direction of both technology and pedagogy: AI systems are becoming ubiquitous and need security, and active learning is proven to be effective. By marrying these, we equip students with relevant skills in an engaging way. The implications are positive for evolving cybersecurity curricula and preparing students for the challenges of securing AI-driven future. We hope that our experience encourages more educators to incorporate AI security content and use or adapt a RAG lab framework for their own teaching.

6 Conclusion

The key contributions of this work are: (1) demonstrating that LLM security concepts can be effectively taught through interactive experimentation, (2) providing a concrete example of a RAG-based educational lab setup that other instructors can replicate or build upon.

Through a two-week module involving attacking and defending a RAG chatbot, students learned about prompt injection attacks, data leakage risks, and mitigation strategies in a tangible way. Our observations and assessments showed that students not only grasped specific vulnerabilities and countermeasures but also developed an improved general security acumen regarding AI systems.

The intersection of AI and security is likely to grow as both a field of research and an area of professional practice. By instilling knowledge and skills in students now, through innovative educational methods, we set the stage for a future where AI-driven systems can be harnessed safely and securely. We believe that hands-on learning, exemplified by the RAG chatbot labs, will be a cornerstone of that educational mission. Our future work will aim to refine and expand these teaching techniques, and we invite the academic community to build upon our findings.

References

[1] L. Ammann and S. Ott. *Analysis of Risks and Mitigation Strategies in RAG - A Framework for Comprehensive Assessment*. 2024.

[2] S. Barnett et al. *Seven failure points when engineering a retrieval augmented generation system*. 2024.

[3] Cloud Security Alliance. *Mitigating Security Risks in Retrieval Augmented Generation (RAG) LLM Applications*. https://cloudsecurityalliance.org/blog/2023/11/22/mitigating-security-risks-in-retrieval-augmented-generation-rag-llm-applications. Accessed 10 February 2025. 2023.

[4] L. Derczynski et al. *garak: A Framework for Security Probing Large Language Models*. https://arxiv.org/abs/2406.11036. 2024.

[5] M. Erickson and P. Kim. "Designing cybersecurity curriculum: Exploring the need for industry certifications and experiential learning". In: *International Association for Computer Information Systems* 22.4 (2021), pp. 9–20.

[6] W. Fan et al. "A Survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models". In: *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. Barcelona, Spain, 2024.

[7] D. Fleischer et al. *RAG Foundry: A Framework for Enhancing LLMs for Retrieval Augmented Generation*. https://arxiv.org/abs/2408.02545. 2024.

[8] Y. Gao et al. *Retrieval-Augmented Generation for Large Language Models: A Survey*. 2024.

[9] W. P. Kelly. "Basic Document Retrieval with Retrieval-Augmented Generation through UIUC.chat". In: *CCSC 38th Annual Southeastern Conference*. Greenville, SC, 2024.

[10] P. Lewis et al. "Retrieval-augmented generation for knowledge-intensive nlp tasks". In: *Advances in Neural Information Processing Systems*. Vol. 33. 2020, pp. 9459–9474.

[11] M. Mukherjee et al. "Strategic Approaches to Cybersecurity Learning: A Study of Educational Models and Outcomes". In: *Information* 15.2 (2024).

[12] J. Ndueso et al. "Cybersecurity Education: The Skills Gap, Hurdle". In: *Innovations in Cybersecurity Education*. Springer International Publishing, 2020, pp. 361–376.

[13] OWASP. *OWASP Top 10 for LLM Applications 2025.* https://owasp.org/www-project-top-10-for-large-language-model-applications/. Accessed 27 February 2025. 2025.

[14] D. Thüs, S. Malone, and R. Brünken. "Exploring generative AI in higher education: a RAG system to enhance student engagement with scientific literature". In: *Frontiers in Psychology* 15 (2024), p. 1474892.

[15] G. Towhidi and J. Pridmore. "Aligning Cybersecurity in Higher Education with Industry Needs". In: *Journal of Information Systems Education* 34.1 (2023), pp. 70–83.

[16] A. Vassilev et al. *Adversarial machine learning: A taxonomy and terminology of attacks and mitigations.* Tech. rep. National Institute of Standards and Technology, 2024.

[17] X. Wang et al. "Searching for Best Practices in Retrieval-Augmented Generation". In: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing.* Miami, Florida, 2024.

[18] C. Wei et al. *Hidden in Plain Sight: Exploring Chat History Tampering in Interactive Language Models.* https://arxiv.org/abs/2404.xxxxx. 2024.

[19] J. Yang, L. Wang, and Y. Wang. "Enhance Student Learning Experience in Cybersecurity Education by Designing Hands-on Labs on Stepping-stone Intrusion Detection". In: *Advances in Science, Technology and Engineering Systems Journal* 6.4 (2021), pp. 355–367.

[20] Y. Yao et al. "A Survey on Large Language Model (LLM) Security and Privacy: The Good, The Bad, and The Ugly". In: *High-Confidence Computing* 4.2 (2024), pp. 2667–2952.

Reviewers — 2025 CCSC Mid West Conference

Imad Al Saeed .... Saint Xavier University, Orland Park, IL
Abbas Attarwala .... California State University, Chico, CA
David Bunde .... Knox College, Galesburg, IL
Jennifer Jo Coy .... Ball State University, Muncie, IN
Khaled Elzayyat .... Salt Lake Community College, Salt Lake, UT
David Furcy .... University of Wisconsin Oshkosh, Oshkosh, WI
Paul Gestwicki .... Ball State University, Muncie, IN
Hector Hernandez .... Benedictine University, Waukegan, IL
Deborah Hwang .... University of Evansville, Evansville, IN
Zachary Kurmas .... Grand Valley State University, Grand Rapids, MI
David Largent .... Ball State University, Muncie, IN
Sugandha Malviya .... Ball State University, Muncie, IN
Rangarajan Parthasarathy .. University of Wisconsin Green Bay, Green Bay, WI
Ann Rangarajan .... Illinois Institute of Technology, Chicago, IL
Michael Rogers .... University of Wisconsin Oshkosh, Oshkosh, WI
Dave Surma .... Indiana University South Bend, South Bend, IN
Paul Talaga .... University of Indianapolis, Indianapolis, IN
James Vanderhyde .... Saint Xavier University, Chicago, IL
Henry Walker .... Grinnell College, Grinnell, IA
Fadi Wedyan .... Lewis University, Romeoville, IL