

The Journal of Computing Sciences in Colleges

**Papers of the 19th Annual CCSC
MidSouth Conference**

April 10th and April 11th, 2026
Lipscomb University
Nashville, TN.

Abbas Attarwala, Editor
California State University, Chico

Kriti Chauhan, Regional Editor
Austin Peay State University

Volume 41, Number 10

April 2026

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.

Copyright ©2026 by the Consortium for Computing Sciences in Colleges. Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the CCSC copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Consortium for Computing Sciences in Colleges. To copy otherwise, or to republish, requires a fee and/or specific permission.

Table of Contents

The Consortium for Computing Sciences in Colleges Board of Directors	7
CCSC National Partners	9
Regional Committees — 2026 CCSC MidSouth Region	10
Serving Interactions: A Social Robot Modeled on Bartender Behavior	11
<i>Jackson Holt, Transylvania University; Kenneth Moorman, Transylvania University</i>	
The Cohort Effect: Enhancing Support and Sense of Belonging for Women of Color in Computing	13
<i>Julia S. Koovakada, Georgia State University; Nannette P. Napier, Georgia State University; Anu G. Bourgeois, Georgia State University</i>	
Usage of Public Computers, Thin Clients & Laptop Dock Stations in an Academic Library	15
<i>Suzanna K. Yaukey, Towson University</i>	
Computational Approaches using Machine Learning to Predict Functional Impact of SERPINA1 Missense Mutations	17
<i>Lucas Hasting, University of North Alabama; Joseph Pope, University of North Alabama; Cynthia Stenger, University of North Alabama; Mark Terwilliger, University of North Alabama</i>	
Proactive Mental Health Assistance via Agentic LLM Chatbots with Retrieval-Augmented Generation	18
<i>Shaira Wajiha, Arkansas Tech University; Robin Ghosh, Arkansas Tech University</i>	
TUI vs GUI: Impact of UI in CS1 Assignments	19
<i>Alexander Ayers, University of West Georgia; Jonathan Corley, University of West Georgia</i>	
Applying Black Box Specifications to Robotics and Bare-Metal Systems	21
<i>Gabriel J. Ferrer, Hendrix College</i>	

Data-Driven Energy Optimization in Smart Steel Manufacturing Using PySpark	23
<i>Haiming Li, Austin Peay State University; Kriti Chauhan, Austin Peay State University</i>	
Design Considerations for Quantum Computing Education Programs	25
<i>Zachary Lloyd, University of Tennessee at Chattanooga; Mani S. Kothapalli, University of Tennessee at Chattanooga; Noah Dillon, University of Tennessee at Chattanooga; Mengjun Xie, University of Tennessee at Chattanooga</i>	
Reflections on Using LEGO Blocks to Build Early Intuition for Vector Spaces in Applied Linear Algebra	26
<i>Abbas Attarwala Ph.D., California State University, Chico; Ed Lindoo Ph.D., Regis University</i>	
Discussions of NP-Complete Problems: An Academic Website	42
<i>Sean McCulloch, Ohio Wesleyan University</i>	
An Explainable Artificial Intelligence Framework for Personalized Diabetes Risk Stratification	52
<i>Shruti Bhandari, Arkansas Tech University; Sanjog Dhakal, Arkansas Tech University; Robin Ghosh, Arkansas Tech University; Tolga Ensari, Arkansas Tech University</i>	
The Impact of Early College and Dual-Enrollment Courses on Success in Upper Level Undergraduate Computer Science Courses	63
<i>James A. Jerkins, University of North Alabama; Daniel Ray, University of North Alabama</i>	
Soccer Action Detection Using TimeSformer: A Transformer-Based Approach to Video Understanding	72
<i>Musfikur Rahaman, University of Arkansas at Little Rock; Robin Ghosh, Arkansas Tech University; Shucheta Bari, University of Arkansas at Little Rock; Abdullah Al Faysal, eTek IT Services Inc.</i>	
Designing a Hardware-Centric Cross-Disciplinary Course in IoT: A Case Study	82
<i>David L. Tarnoff, East Tennessee State University</i>	

From Points to Proficiency: How Students Perceive Standards-Based Grading in Computing Courses	92
<i>Jonathan McCurdy, Mount St. Mary's University; Ruth Lamprecht, Mount St. Mary's University</i>	
Scaffolded and Culturally Relevant Python Learning with AI as a Socratic Partner in K–12 Computing	102
<i>Marlon Mejias, University of North Carolina at Charlotte; Oluwagbeminiyi Johnson, University of North Carolina at Charlotte; Abi Olukeye, University of North Carolina at Charlotte; Luce-Melissa Kouaho, University of North Carolina at Charlotte; Victor Mack, University of North Carolina at Charlotte</i>	
A Comprehensive Survey of Agentic AI: Design Principles, Security Risks, and Ethical Considerations	112
<i>Md Shaba Sayeed, Arkansas Tech University; Robin Ghosh, Arkansas Tech University; MD Abdus Salam Siddique, Arkansas Tech University; A K M Rubaiyat Reza Habib, Arkansas Tech University</i>	
FedRAG: A Federated Retrieval-Augmented Generation System with Advanced Privacy and Efficiency	127
<i>Kesava Manikanta Chirumamilla, Arkansas Tech University; Md Abdus Salam Siddique, Arkansas Tech University</i>	
Behavioral Authentication using Password String Characteristics and Machine Learning Techniques	136
<i>Dylan Lott, University of South Carolina Aiken; Brandon Weathers, University of South Carolina Aiken; Yilian Zhang, University of South Carolina Aiken</i>	
Learning Cryptography One Tap at a Time: An NFC-Based Payment System for the Classroom	145
<i>Bilal Shebaro, St. Edward's University</i>	
Comparison of Software Development and Communication Skills by Degree Options	154
<i>Karern Works, Florida State University Panama City; Sharanya Jayaraman, Florida State University</i>	
Reviewers — 2026 CCSC MidSouth Conference	164

The Consortium for Computing Sciences in Colleges

Board of Directors

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

Bryan Dixon, President (2026), bcdixon@csuchico.edu, Computer Science Department, California State University Chico, Chico, CA 95929.

Shereen Khoja, Vice President/President-Elect (2026), shereen@pacificu.edu, Computer Science, Pacific University, Forest Grove, OR 97116.

Abbas Attarwala, Publications Chair (2027), aattarwala@csuchico.edu, Department of Computer Science, California State University Chico, Chico, CA 95929.

Ed Lindoo, Treasurer (2026), elindoo@regis.edu, Anderson College of Business and Computing, Regis University, Denver, CO 80221.

Haiyan Cheng, Membership Secretary (2028), hcheng@willamette.edu, Department of Computer Science, Willamette University, Salem, OR 97301.

Judy Mullins, Central Plains Representative (2026), mullinsj@umkc.edu, University of Missouri-Kansas City, Kansas City, MO (retired).

Michael Flinn, Eastern Representative (2026), mflinn@frostburg.edu, Department of Computer Science & Information Technologies, Frostburg State University, Frostburg, MD 21532.

Gabe Ferrer, Midsouth Representative (2028), ferrer@hendrix.edu, Department

of Computer Science, Hendrix College, Conway, AR 72032.

David Largent, Midwest Representative (2026), dllargent@bsu.edu, Department of Computer Science, Ball State University, Muncie, IN 47306.

Michael Gousie, Northeastern Representative (2028), gousie_mike@wheatoncollege.edu, Computer Science Department, Wheaton College, Norton, MA 02766.

Ben Tribelhorn, Northwestern Representative (2027), tribelhb@up.edu, School of Engineering, University of Portland, Portland, OR 97203.

Mohamed Lotfy, Rocky Mountain Representative (2028), MohamedL@uvu.edu, Information Systems & Technology Department, College of Engineering & Technology, Utah Valley University, Orem, UT 84058.

Mika Morgan, South Central Representative (2027), mika.morgan@msutexas.edu, Department of Computer Science, Midwestern State University, Wichita Falls, TX 76308.

Karen Works, Southeastern Representative (2027), keworks@pc.fsu.edu, Department of Computer Science, Florida State University Panama City, Panama City, FL 32405.

Michael Shindler, Southwestern Representative (2026), mikes@uci.edu, Computer Science Department, UC Irvine, Irvine, CA 92697.

Serving the CCSC: These members are serving in positions as indicated:

Bin Peng, Associate Editor, bin.peng@park.edu, Computer Science and Information Systems, Park University, Parkville, MO 64152.

Lucas Cordova, Associate Treasurer, lpcordova@willamette.edu, Department of Computer Science, Willamette University, Salem, OR 97301.

George Dimitoglou, Comptroller, dimitoglou@hood.edu, Department of Computer Science, Hood College, Frederick, MD 21701.

Megan Thomas, Membership System

Administrator, mthomas@cs.csustan.edu, Department of Computer Science, California State University Stanislaus, Turlock, CA 95382.

Karina Assiter, National Partners Chair, KarinaAssiter@landmark.edu, Landmark College, Putney, VT 05346.

Ed Lindoo, UPE Liaison, elindoo@regis.edu, Anderson College of Business and Computing, Regis University, Denver, CO 80221.

Deborah Hwang, Webmaster, hwangdjh@acm.org.

2026 CCSC Midsouth Sponsors/Partners

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

Sponsors/Partners

ACM2y & ACM CCECC

Blossoms

College Board

Eduicator

MAP-CS

Rephactor

2026 CCSC Midsouth Steering Committee

Gabe Ferrer *Regional Representative, Hendrix College, AR*
Kriti Chauhan *Regional Editor, Austin Peay State University, TN*
Gabe Ferrer *Registrar, Hendrix College, AR*
Ed Lindoo *Treasurer, Regis University, CO*
Mark Goadrich *Webmaster, Hendrix College, AR*

2026 CCSC Midsouth Conference Committee

Susan Hammond *Conference Chair, Lipscomb University, TN*
Robin Ghosh *Papers Chair, Arkansas Tech University, AR*
Kriti Chauhan *Posters Chair, Austin Peay State University, TN*
Olivera Grujic *Workshops/Tutorials/Panels Chair, California State University – Stanislaus, CA*
Mario Guimares . *Exhibitors Chair, Southeast Missouri State University, MO*
Rao Li *Publicity Chair, University of South Carolina – Aiken, SC*

Serving Interactions: A Social Robot Modeled on Bartender Behavior*

Poster Abstract

Jackson Holt, Kenneth Moorman
Department of Computer Science
Transylvania University
Lexington, KY 40508
`{jrholt26, kmoorman}@transy.edu`

Human social interactions are inherently complex, relying on a combination of verbal and nonverbal cues to convey and interpret emotional states. In everyday exchanges, subtle signals such as facial expressions and vocal tone allow humans to recognize and respond to one another's emotions with remarkable nuance. The question becomes more challenging when attempting to equip machines with similar capabilities. This projects goal was to study the effectiveness of implementing emotion recognition systems with the Furhat robot [1].

Facial emotion recognition was implemented using the DeepFace library [2], which analyzes facial features through a multi-stage process including detection, alignment, feature extraction, and emotion classification based on the breakdown of the face[3]. In addition, speech sentiment was analyzed using the Vader sentiment analyzer to classify conversational input as positive, negative, or neutral. Combining visual and linguistic signals allowed the robot to form a more robust assessment of a user's emotional state.

To promote natural interaction, emotional responses were reduced to three core states: positive, negative, and neutral. enabling smoother conversational transitions. The system was modeled after brief, casual social exchanges, such as those between a bartender and customers [4] emphasizing short but contextually appropriate dialogue.

*Copyright is held by the author/owner.

References

- [1] *Furhat Robotics* — *furhatrobotics.com*. <https://www.furhatrobotics.com/>. [Accessed 11-08-2025].
- [2] Ramachandran Venkatesan et al. “Human emotion detection using DeepFace and artificial intelligence”. In: *Engineering Proceedings* 59.1 (2023), p. 37.
- [3] Sandeep Jain. *Deep Face Recognition - GeeksforGeeks* — *geeksforgeeks.org*. <https://www.geeksforgeeks.org/machine-learning/deep-face-recognition/>. [Accessed 07-08-2025]. 2022.
- [4] Eight Six. *How to Keep your Guests Engaged: 5 Ideas for Starting Conversation at your Bar*. <https://abarabove.com/engaging-guests-conversation/>. [Accessed 07-08-2025]. 2019.

The Cohort Effect: Enhancing Support and Sense of Belonging for Women of Color in Computing*

Poster Abstract

Julia S. Koovakada¹, Nannette P. Napier¹, Anu G. Bourgeois²

¹Department of Computer Information Systems

²Department of Computer Science

Georgia State University

Atlanta, GA 30303

{jkoovakada2, nnapier1, abourgeois}@gsu.edu

Undergraduate computing students from historically underrepresented groups often report a low sense of belonging, which can negatively affect persistence in technical programs [1, 2, 3]. Women in computing, especially women of color, may face barriers due to limited representation and a perceived unwelcoming culture in undergraduate computing programs [4, 5].

To address these challenges, the RISE in Computing Scholars program is a semester-long, cohort-based initiative designed to broaden participation and foster a sense of belonging for students from historically underrepresented groups in computing. The program emphasizes improving the experiences and outcomes of women in computing while promoting a more inclusive culture through professional development workshops and structured mentoring.

Prior research from the program’s first year identified increased sense of belonging among Black women participants through participation in the program [6]. This study expands our focus to women of color across all four program semesters from 2023 to 2025. We analyzed pre- and post-surveys of 102 women of color participants to address the following research question: To what extent does the RISE in Computing Scholars program increase women of color’s sense of belonging (SoB), and which support dimensions—departmental climate, peer networks, or mentorship relationships—show the strongest gains?

*Copyright is held by the author/owner.

Paired-samples t-tests revealed a statistically significant increase in participants' perceived sense of belonging, with post-program scores averaging 0.48 points higher than pre-program scores. Significant gains were also observed in perceived mentor support ($\Delta = 0.88$) and peer support ($\Delta = 0.51$), suggesting that interpersonal support systems play a critical role in fostering belonging among women of color in computing. In contrast, perceived departmental support exhibited a small but statistically significant decrease of 0.22 points, highlighting persistent structural or climate-level challenges that may extend beyond the scope of cohort-based interventions. Future work will further examine which components of support most strongly contribute to students' sense of belonging.

References

- [1] S. Mithun and X. Luo. "Improving Perceptions of Underrepresented Students towards Computing Majors through Mentoring". In: *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1*. 2023, pp. 26–32. DOI: 10.1145/3587102.3588817.
- [2] V. Ojha, L. West, and C. M. Lewis. "Computing self-efficacy in undergraduate students: A multi-institutional and intersectional analysis". In: *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*. 2024, pp. 993–999. DOI: 10.1145/3626252.3630811.
- [3] S. R. Fisk et al. "Retaining Black Women in Computing: A Comparative Analysis of Interventions for Computing Persistence". In: *ACM Trans. Comput. Educ.* 24.2 (2024). Article 20. DOI: 10.1145/3635313.
- [4] A. Solomon et al. "Not just Black and not just a woman: Black women belonging in computing". In: *2018 Research on Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT)*. IEEE, 2018, pp. 1–5. DOI: 10.1109/RESPECT.2018.8491700.
- [5] Y. A. Rankin, J. O. Thomas, and S. Erete. "Black women speak: Examining power, privilege, and identity in CS education". In: *ACM Transactions on Computing Education (TOCE)* 21.4 (2021), pp. 1–31. DOI: 10.1145/3451344.
- [6] Nannette P. Napier and Anu G. Bourgeois. "RISE in Computing Scholars: A Cohort-Based Community for Black Women". In: *Proceedings of the 2024 RESPECT Annual Conference*. 2024, pp. 43–50. DOI: 10.1145/3653666.3656100.

Usage of Public Computers, Thin Clients & Laptop Dock Stations in an Academic Library*

Poster Abstract

Yaukey, Suzanna K.
Albert S. Cook University Library
Towson University
Towson, MD 21252
syaukey@towson.edu

For decades, academic libraries have often housed the largest public computer labs on campuses[1][2], but student expectations may be changing. Per Stvilia and Gibradze, 59% of students choose to use the library for its information and computer resources, second only to studying[3]. Elkins et al. indicate that nearly 60% of students used a campus computer at least once a week[4]. Chao et al. find that 74% of students claim that they always bring a laptop to the library[5].

With more and more students bringing their own computers and laptops to campus, virtualization can be one answer to stretching budget funds effectively and sustainably, allowing students to access software solutions without installing additional software applications. Thin clients are often more sustainable than traditional hardware with preloaded software[6], but they rely on servers for "processing power, memory, data storage and all sorts of application software"[7].

This study includes a qualitative analysis of observation and interview data conducted with students using technology on one floor of an academic library. Contributing this knowledge, even if from a small subset of students and patrons, could help many libraries, and even academic public computing labs, consider different options in the future. Observations supported the grouping of three types of students: (1) the solitary student, (2) the collaborative pair or trio, and (3) the social pair. Students used various types of software on the library computers and completed academic, social, and personal activities. During interviews, two further themes emerged: the importance of

*Copyright is held by the author/owner.

ambiance to students when selecting study spaces and the helpfulness of technologies such as monitors, connection cables that charge, access to software, etc. for students. Furthermore, unexpected affordability and communication issues emerged during interview discussions.

References

- [1] Krista Graham. “When the Library Becomes the Largest Computer Lab on Campus: Supporting Productivity Software in an Academic Environment”. In: *College & Research Libraries News* 64.7 (July 2003), pp. 462–68.
- [2] Sheryl Moore. “Libraries to Labs: Managing Public Access Computer Labs in an Academic Library Environment”. In: *The Reference Librarian* 35.74 (2001), pp. 207–20.
- [3] Besiki Stvilia and Leila Gibradze. “Examining Undergraduate Students’ Priorities for Academic Library Services and Social Media Communication”. In: *The Journal of Academic Librarianship* 43.3 (May 2017), pp. 257–262.
- [4] Susan Elkins et al. “What Do You Want from Us? Evaluating Student Interest in Technology-Based Services in Academic Libraries”. In: *College & Research Libraries* 81.5 (2020), pp. 844–859. DOI: <http://dx.doi.org/10.5860/crl.81.5.844>.
- [5] Zao Chao et al. “Understanding User Experience in Bring Your Own Device Spaces in the Library: A Case Study of Space Planning and Use at a Large Research University”. In: *Performance Measurement and Metrics* 20.3 (2019), pp. 201–212. DOI: <http://dx.doi.org/10.1108/PMM-07-2019-0027>.
- [6] Jesús Martínez-Mateo, Susana Muñoz-Hernandez, and David Pérez-Rey. *A Discussion of Thin Client Technology for Computer Labs*. Preprint. 2010. DOI: 10.48550/arxiv.1005.2534. arXiv: 1005.2534.
- [7] Sumita U. Sharma and Y.B. Gandole. “Virtualization Approach to Reduce Network Latency for Thin Client Performance Optimization in Cloud Computing Environment”. In: *2014 International Conference on Computer Communication and Informatics*. Coimbatore, India: IEEE, 2014, pp. 1–6. DOI: <http://dx.doi.org/10.1109/ICCCI.2014.6921753>.

Computational Approaches using Machine Learning to Predict Functional Impact of SERPINA1 Missense Mutations*

Poster Abstract

Lucas Hasting¹, Joseph Pope²,
Cynthia Stenger², Mark Terwilliger¹

¹Department of Computer Science and Information Systems

²Department of Mathematics

University of North Alabama

Florence, AL 35632

{lucashasting03, jpope20jr}@gmail.com

{clstenger, mterwilliger}@una.edu

Alpha-1 antitrypsin (AAT) is a protease inhibitor made in the liver with the goal of protecting the lungs from irritants. The lack of AAT in the lungs may make them more easily damaged by smoking or through pollution and dust in the air. An AAT deficiency is caused by mutations in the SERPINA1 gene, the gene that carries the instruction for making the AAT protein.

This study used bioinformatics and machine-learning approaches to investigate the pathogenic potential of missense variants in the SERPINA1 gene associated with alpha-1 antitrypsin (AAT) deficiency. Specifically, we studied whether pathogenicity can be predicted in variants associated with SERPINA1 and AAT.

Four models were developed and evaluated: a Multi-Layer Perceptron (MLP), a Decision Tree, a Random Forest, and a K-Nearest Neighbor (K-NN) classifier. These models achieved accuracies of 86%, 89%, 90%, and 85%, respectively, in determining whether a given Variant of Uncertain Significance (VUS) was benign or pathogenic.

*Copyright is held by the author/owner.

Proactive Mental Health Assistance via Agentic LLM Chatbots with Retrieval-Augmented Generation*

Poster Abstract

Shaira Wajiha, Robin Ghosh

Department of Engineering and Computer Sciences

Arkansas Tech University

Russellville, AR 72801

{swajiha, rghosh}@atu.edu

Mental health challenges such as anxiety and depression are widespread, yet often remain unaddressed due to stigma, financial barriers, limited access to professionals, or personal reluctance. To address this gap, we present an accessible AI-powered chatbot that provides preliminary conversational support with empathetic, context-aware responses. The chatbot is trained and evaluated on two publicly available counseling conversation datasets from Huggingface, enabling it to learn and maintain therapeutic dialogue patterns. We compare three advanced LLMs, such as Llama3.1, Mistral 7b, and Qwen3, evaluating them on relevance, empathy, conciseness, and contextual understanding, and all models demonstrate high response quality. Incorporation of a Retrieval-Augmented Generation (RAG) pipeline with Facebook AI Similarity Search (FAISS) enhances factual accuracy and contextual relevance while reducing hallucinations. A key innovation is the integration of agentic AI, empowering the chatbot to recognize extreme distress and, with user consent, assist in scheduling appointments with mental health professionals. Overall, our system demonstrates the potential to deliver reliable, empathetic, and actionable conversational AI support for individuals experiencing early-stage mental health distress.

Keywords: Mental health, Conversational AI, LLM, RAG, FAISS, Agentic AI.

*Copyright is held by the author/owner.

TUI vs GUI: Impact of UI in CS1 Assignments*

Poster Abstract

Alexander Ayers, Jonathan Corley
School of Computing, Analytics, and Modeling
University of West Georgia
Carrollton, GA 30118

alexayers2000@gmail.com, jcorley@westga.edu

CS1 is practical and hands-on in nature with students typically completing multiple coding assignments that put into practice the concepts learned during the course. There are several common formats for CS1 coding assignments with a distinction commonly drawn between assignments that use a graphical user interface (GUI) and assignments with a text-based interface (TUI). Prior studies have shown that graphical assignments are more engaging to students, whereas console assignments provide a less complex structure [1].

This study examines the possible effect of assignment format on student learning, examining concept comprehension, number and type of errors, and assignment completion. The type of errors are categorized into syntactic errors or logical errors, and learning outcomes are compared via a post-assignment quiz. The study participants are students from an objects-first CS1 course in Java. During the course workshops, students complete a coding assignment during the first workshop day of the week followed by a conceptual quiz during the following workshop day. There were 35 students in the TUI group and 29 students in the GUI group.

The best comparison to this study, subject-wise, is the work of Ball *et al.* which focuses on evaluating a whole course with groups of students swapping assignment types consistently throughout the semester [2]. The study focused on student preference towards one format or the other. Surveys performed by Ball *et al.* found that TUIs were preferred for the perceived simplicity compared to the GUI alternatives [2]. Alphonse and Ventura note that GUIs were preferred for type of project interface due to students' familiarity with GUIs in their day-to-day lives, as well as students' desire to improve their

*Copyright is held by the author/owner.

skills [1]. However, despite noting a similar GUI preference from students, Ball *et al.* found that with TUI assignments students performed better, enjoyed the assignment more, and spent less time on the assignment [2].

Our study results present a statistically significant increase in logical errors when assignments include implementing a project using a GUI as opposed to a TUI, but no difference in syntax errors or learning outcomes was observed. Therefore, we recommend the simpler TUI interface unless instructors intend for students to grapple with the more complex system structure of a GUI understanding this may increase the occurrence of logical errors. Future work could replicate this study in a different context and examine if student perception of the assignment differs across assignment format (GUI v TUI). It would also be of interest to study if our results hold when the assignment is offered as a homework or project rather than during a time-constrained lab environment.

References

- [1] Carl Alphonse and Phil Ventura. “Using Graphics to Support the Teaching of Fundamental Object-Oriented Principles in CS1”. In: *Companion of the 18th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications*. OOPSLA ’03. Anaheim, CA, USA: Association for Computing Machinery, 2003, pp. 156–161. ISBN: 1581137516.
- [2] Robert Ball et al. “GUI-Based vs. Text-Based Assignments in CS1”. In: *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*. SIGCSE ’18. Baltimore, Maryland, USA: Association for Computing Machinery, 2018, pp. 1017–1022. ISBN: 9781450351034.

Applying Black Box Specifications to Robotics and Bare-Metal Systems*

Poster Abstract

Gabriel J. Ferrer

Department of Mathematics and Computer Science

Hendrix College

Conway, Arkansas 72032

`ferrer@hendrix.edu`

Harlan Mills introduced black box specifications to enable software engineers to formally verify that every possible input scenario has a well-defined response. We are studying black box specification for implementing robot controllers using ROS2 in a robotics course and implementing bare-metal code in an operating systems course.

Black box specifications model how a software artifact interacts with its environment without explicitly representing its internal state. Each input from the environment is termed a stimulus and each output is termed a response. A history is a finite-length sequence of alternating stimuli and responses. A well-formed black box specification is a function from the domain of the set of history-stimulus pairs to the range of the set of responses.

In ROS2, each unit of robot control is termed a node. Each node sends information and requests to other nodes by publishing messages and receives information and requests from other nodes by subscribing to messages. Each message received via subscription (or direct hardware input) is a stimulus and each published message (or direct hardware output) is a response. By building a black-box specification for a node in this manner, we can formally verify that every message received produces the intended observable behavior.

The bare-metal environment in which our students develop software in the operating systems course allows for keyboard input and output in the x86 VGA buffer. Each keystroke constitutes a stimulus and each character displayed on the monitor constitutes a response. Again, by building a black-box specifica-

*Copyright is held by the author/owner.

tion for the application, we can formally verify that each keystroke received produces the intended observable behavior.

Having formulated this approach, we plan to apply it in the coming academic year in these two courses and assess its utility in guiding students towards producing software that matches their intentions in these domains.

Data-Driven Energy Optimization in Smart Steel Manufacturing Using PySpark*

Poster Abstract

Haiming Li, Kriti Chauhan

Department of Computer Science and Information Technology

Austin Peay State University

Clarksville, TN 37044

hli2@students.apsu.edu, chauhank@apsu.edu

Energy consumption represents a significant operational cost and environmental footprint in heavy manufacturing, particularly within the steel industry. As facilities transition to “smart” monitoring systems, the volume of sensor data presents an opportunity to optimize usage through predictive modeling. This study analyzed 35,040 instances of operational data from a South Korean steel mill [1], sourced via the UCI Machine Learning Repository [2], using a PySpark ML pipeline to forecast demand and identify consumption drivers.

Exploratory analysis revealed risk of data leakage via the derived metric Carbon Dioxide (CO_2), which was subsequently excluded to ensure model robustness. The study formulated the hypothesis that inductive loads, represented by Lagging Current Reactive Power [3], serve as the dominant predictor of active energy consumption (*Usage_kWh*). The resulting Linear Regression model achieved a Root Mean Squared Error (RMSE) of 9.99 kWh and an R^2 of 0.91, validating the hypothesis.

Analysis of regression coefficients confirmed that Lagging Current Reactive Power is the most significant variable (coefficient ≈ 1.48), vastly outweighing temporal features such as time-of-day. These findings suggest that energy optimization strategies in this context should prioritize hardware solutions, such as the installation of capacitor banks to correct power factor inefficiencies [4]. This project demonstrates the efficacy of Spark for industrial data analysis and provides actionable insights for manufacturing environments.

*Copyright is held by the author/owner.

Keywords: PySpark, Linear Regression, Energy Optimization, Smart Manufacturing.

References

- [1] V. E. Sathishkumar, Changsun Shin, and Yongyun Cho. “Efficient energy consumption prediction model for a data analytic-enabled industry building in a smart city”. In: *Building Research & Information* 49.1 (2021), pp. 127–143.
- [2] UCI Machine Learning Repository. *Steel Industry Energy Consumption Dataset*. 2021. URL: <https://archive.ics.uci.edu/ml/datasets/Steel+Industry+Energy+Consumption+Dataset>.
- [3] IEEE Standard 141-1993. *Recommended Practice for Electric Power Distribution for Industrial Plants*. Tech. rep. IEEE, 1993.
- [4] U.S. Department of Energy. *Reducing Power Factor Cost*. Tech. rep. Office of Energy Efficiency & Renewable Energy, 2014.

Design Considerations for Quantum Computing Education Programs*

Poster Abstract

Zachary Lloyd, Mani S. Kothapalli, Noah Dillon, Mengjun Xie
Department of Computer Science and Engineering
University of Tennessee at Chattanooga
Chattanooga, TN 37403

zachary-lloyd@utc.edu, {qkj195, pqf696}@mocs.utc.edu
mengjun-xie@utc.edu

Quantum Computing is progressing from theory to practical application. However, not all problems can be effectively adapted to quantum systems. Navigating this disparity requires a deep understanding of the application areas from experts in those fields who also have a sufficient understanding of quantum systems.

The goal of this work is to provide practical guidance for designing Quantum Computing education programs that reach a wider range of learners. Many potential students do not have the same backgrounds as the Quantum Computing experts they will be working alongside. Simply opening more seats in traditional courses and using the same mathematically rigorous instruction methods will not address this gap.

This poster presents an initial set of program design considerations, focusing on when quantum topics should be introduced, who should be the focus, and what topics should be taught. These considerations are based on principles from curriculum design, interdisciplinary learning theory, and observed challenges among non-STEM learners. The poster also provides suggestions, tools, and resources to help educators develop curricula.

This work was supported in part by the National Science Foundation (Award 2234910), the Department of War Cyber Service Academy Scholarship Program (Award HQ00342510050), and the University of Tennessee at Chattanooga (CEACSE Award R041302314).

*Copyright is held by the author/owner.

Reflections on Using LEGO Blocks to Build Early Intuition for Vector Spaces in Applied Linear Algebra*

Abbas Attarwala Ph.D.¹ and Ed Lindoo Ph.D.²

¹Computer Science Department
California State University
Chico, CA, 95973

aattarwala@csuchico.edu

²Computer Information Systems
Anderson School of Business

Regis University
Denver, CO, 80221

elindoo@regis.edu

Abstract

This paper presents a personal reflection on a classroom activity developed and used in an applied linear algebra course within a computer science degree program to support early intuition-building around vector spaces. Students typically encounter vector spaces through formal definitions and axiom verification, including properties such as closure under vector addition and scalar multiplication, associativity and commutativity of addition, the existence of a zero vector and additive inverses, and the distributive properties of scalar multiplication. Although this approach establishes correctness, it often leaves students with little sense of why the axioms matter or what they imply about vector addition. In this paper, we present an activity that uses 2×2 LEGO blocks as tangible stand-ins for abstract vectors and is designed for use early in

*Copyright ©2026 by the Consortium for Computing Sciences in Colleges. Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the CCSC copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Consortium for Computing Sciences in Colleges. To copy otherwise, or to republish, requires a fee and/or specific permission.

the semester, before formal definitions are introduced. By working with physical 2×2 LEGO blocks rather than arrows, the activity avoids reinforcing the assumption that vectors are inherently geometric and helps prepare students to later interpret vectors as elements of diverse spaces, including spaces of random variables, functions, and polynomials commonly encountered in computer science applications.

1 Introduction

Vector spaces and related linear algebraic structures are foundational in many areas of computer science, underlying topics such as machine learning, data analysis, and randomized algorithms (Dereziński & Mahoney, 2024; Kannan & Vempala, 2017). In applied linear algebra courses for computer science and related STEM students, however, students often encounter vector spaces primarily through symbolic definitions and axioms (Dorier & Sierpinska, 2001; Fernández-Cézar et al., 2020). While this approach is mathematically sound, it can leave students with little intuition for why vector addition is defined the way it is, or how algebraic requirements such as closure and commutativity constrain the structure of the objects being added. As a result, students may treat vector space axioms as formal conditions to verify rather than as principles that shape representation and computation (Berman & Shvartsman, 2016; Trigueros et al., 2024).

This paper presents a personal reflection on a classroom activity developed and used in an applied linear algebra course for computer science students. By working with physical 2×2 LEGO blocks rather than arrows, the activity avoids reinforcing the assumption that vectors must be understood primarily as geometric objects and helps prepare students to interpret vectors as elements of more general spaces, such as spaces of functions, polynomials, or random variables encountered in computer science applications.

Students work within a clearly specified collection of allowable LEGO objects: towers constructed by stacking copies of a 2×2 LEGO blocks block. The number of stacked layers may vary, as shown in Figure 1. Students are first asked to take two such objects and combine them in any way they choose, without restrictions on how the pieces are joined. A constraint is then introduced: the result of the combination must still belong to the original collection; that is, when viewed from above, it must still appear as a single 2×2 LEGO blocks block, although its height may increase or decrease. Under this constraint, only vertical stacking is admissible. This shift from unconstrained to constrained combination focuses attention on which operations preserve membership in the collection, motivating the idea of closure before any formal definition of addition is introduced.

The classroom activity is organized around two contrasting cases. In the

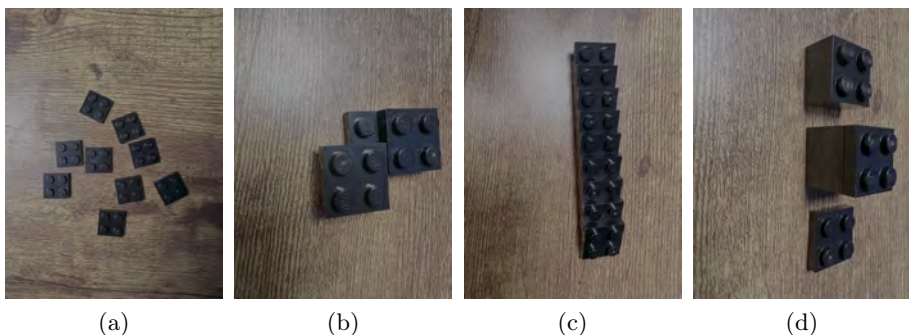


Figure 1: Exploring closure through constrained combination. Viewed from above, (a) shows the initial collection of identical 2×2 LEGO blocks. In (b) and (c), students freely combine elements, producing structures that no longer preserve the 2×2 footprint and thus fall outside the original collection. In (d), imposing the constraint that combinations must preserve the 2×2 footprint restricts admissible operations to vertical stacking, yielding a taller object that remains within the collection.

first, all blocks are the same color. Here, students observe that vertical stacking is the only way to combine towers while preserving membership in the set of allowable objects. Moreover, stacking produces towers that are physically identical regardless of order: stacking two blocks on four yields the same structure as stacking four on two. In this setting, the physical model aligns naturally with the algebraic property of commutativity. In the second case, blocks of multiple colors are introduced. Students observe that stacking four black blocks followed by two blue blocks produces a visibly different tower than stacking two blue blocks followed by four black blocks, despite the expectation that these combinations should represent the same result if addition is commutative. This mismatch between physical representation and algebraic expectation becomes a focal point for discussion. To resolve it, students must decide which aspects of the LEGO structures are mathematically relevant and which can be ignored. Re-describing towers in terms of quantities that do not depend on physical arrangement, specifically the number of blocks of each color present, leads naturally to a coordinate-based representation and motivates component-wise addition. From this perspective, the familiar structure of $\mathbb{R}^2$ emerges as a consequence of enforcing closure and commutativity rather than as a pre-defined algebraic model. By allowing students to experience both alignment and conflict between representation and algebraic structure, the LEGO activity provides a tangible foundation for building intuition about vector spaces in a computer science context.

2 Related Work

Studies such as Stewart and Thomas, 2019 highlight that, while students may reproduce or memorize formal results in linear algebra, they often struggle to understand why those results hold or how proofs give them meaning. A common obstacle is the disconnection between formal definitions and students' intuitive or prior mathematical understanding (Dorier et al., 2000). Students often enter with geometric intuitions that interpret vectors as arrows in $\mathbb{R}^2$ or $\mathbb{R}^3$, and they may struggle to reconcile these intuitions with the general algebraic formulation of vectors as elements of an abstract space (Dorier, 2003).

Moreover, students may have difficulty moving beyond geometric interpretations of vectors to non-standard vector spaces, such as function or matrix spaces, where elements are no longer familiar geometric objects (Dorier, 2003). This difficulty is compounded when instruction and assessment emphasize algorithmic tasks, as students may succeed procedurally while still holding severe misunderstandings about elementary concepts (Dorier et al., 2000). Such misunderstandings can make it difficult for students to reason about properties such as closure or linear independence in unfamiliar contexts. Proof and rigor further exacerbate these challenges. Many students struggle with verifying mathematical properties, constructing justifications, and using counterexamples; more broadly, reasoning and proof tasks often require higher-order thinking that students may not have developed through procedural or rote approaches (Lessing & Ogonnaya, 2025). In linear algebra, this difficulty appears in the leap from algorithmic problem-solving to axiomatic reasoning, such as proving that a proposed structure satisfies the properties required of a vector space.

Recent research has focused on addressing these challenges through alternative teaching methods. One major trend is the shift from definition-first instruction to activity-based and inquiry-driven learning. Fernández-Cézar et al., 2020 implemented a problem-based learning (PBL) approach that introduced vector spaces through real-world engineering scenarios. Students formulated and solved systems of linear equations, which naturally led to the idea of solution spaces. Compared to a control group receiving traditional instruction, the PBL group exhibited reduced pressure and effort while achieving comparable conceptual understanding, along with higher overall course achievement. Cuevas-Vallejo et al., 2023 developed a five-task instructional sequence set in a robotic arm simulation. Through structured exploration with GeoGebra-based virtual interactive didactic scenarios, students transitioned from geometric representations of vectors as arrows to algebraic representations as ordered pairs. More than 70% of participants successfully conceptualized vectors both graphically and algebraically. This work shows how dynamic digital environments can support students in linking algebraic representations with geometric inter-

pretation. Studies by Aytekin and Kiymaz, 2019 and Turgut et al., 2022 report that visualization supports students' reasoning about linear combinations and span, helping them coordinate algebraic representations with geometric structures such as lines and planes.

Several researchers have explored the use of physical and embodied tools to support linear algebra instruction. Beltrán Meneu and Murillo Arcila, 2016 presented a color-based instructional model in which RGB color mixing is used to introduce linear combinations, generating sets, bases, and convexity, with GeoGebra supporting the visualization of these relationships. Cook, 2022 similarly described LEGO-based stop-motion animations for topics such as matrices, vectors, linear geometry, and linear transformations, showing how familiar physical models can be used to make abstract linear algebra concepts more accessible.

The LEGO-based closure-first activity developed in this paper offers a distinct pedagogical contribution. Unlike most innovations that begin with span or linear combinations, our contribution emphasizes foundational axioms of vector spaces particularly closure, commutativity, and identity before introducing bases or coordinates.

Students use colored LEGO bricks to model vectors, with stacking representing vector addition. Each color corresponds to a basis direction, and stacking same-colored bricks enforces component-wise addition. This hands-on approach lets students physically explore and test properties of vector spaces, providing tangible counterexamples when axioms fail. By guiding students to verify closure and other properties before introducing formal definitions, the activity supports proof-readiness and reduces abstraction. The use of color to encode basis elements is another unique feature. It enables students to discover the concept of a minimal generating set through physical experimentation: removing a color (basis vector) restricts what towers (vectors) can be built. This provides a concrete experience of independence and span, helping students understand basis minimality and coordinate representation. Finally, the playful and familiar nature of LEGO reduces student anxiety and encourages collaborative exploration. Compared to other embodied methods, the LEGO activity uniquely foregrounds the axiomatic structure of vector spaces through direct manipulation, offering a novel and intuitive pathway to abstraction.

3 Activity Design and Classroom Implementation

We describe two closely related classroom activities designed to build intuition for vector spaces by engaging students with the vector space axioms prior to formal definition. Both activities use 2×2 LEGO blocks as manipulatives and are structured around explicitly defined collections of allowable objects.

In each activity, the collection consists of LEGO constructions whose top-down footprint is a 2×2 square. No restrictions are placed on height, and no rules for combining objects are given initially. This framing emphasizes that students are working within a single, well-defined collection of objects that are all of the same kind, differing only in the number of stacked layers. It mirrors the mathematical setting of vector spaces, where the elements under consideration must first belong to a common type, such as column vectors, functions, polynomials, or random variables, before questions about how they may be combined are meaningful.

The two activities differ only in the information carried by elements of the collection. In the first activity, all blocks are the same black color, so objects are distinguished solely by height. In the second activity, blocks may be white, black, or blue, and objects may contain arbitrary numbers of each color, making color counts salient. In both activities, students are asked to take two objects from the relevant collection and combine them in any way they choose, subject only to the requirement that the result still be an allowable object in that activity. Through this process, students discover that preserving the footprint constraint forces combinations to proceed by vertical stacking, yielding an operation that satisfies closure and motivates later formal discussion of vector addition.

3.1 Activity 1: A Single Generating Element

In the first activity, students are given a collection of identical black 2×2 LEGO blocks. The set of objects under consideration is defined as all configurations whose top-down view is a 2×2 square, with height allowed to vary. Any construction that violates this footprint constraint is excluded. Within this setting, every object students construct consists of some number of copies of the same block type.

As students explore the collection, they produce configurations of varying heights, including configurations above the table and, by convention, configurations below the table representing negative quantities. When asked to describe these objects aloud, students naturally refer to them in terms of the number of black blocks involved, such as “five black blocks,” “three black blocks,” or “one black block.” Through this discussion, it becomes clear that all objects in the collection are described using the same fundamental unit.

At this point, the instructor highlights that every admissible object in the set is obtained by combining multiple copies of a single building block. If this block is removed from consideration, there is no remaining way to describe or generate the objects in the collection. In linear algebra language, this block plays the role of a generating element: all objects in the set are constructed from it, and none can be constructed without it. This mirrors the role of the

number 1 on the number line, which serves as the reference unit from which all integers are generated.

Because there is only one block type, questions of independence and minimality are trivial in this activity. There are no alternative building blocks, and every object in the set is generated by repeated stacking of the same element. In this sense, the activity establishes a baseline: students experience a one-dimensional setting in which physical structure, description, and generation align naturally. This discussion prepares students to think about structure before operation: what it means for objects to belong to the same set, how they are described, and what minimal information is required to generate them.

With this structure in place, students explore how two objects from the collection can be combined while preserving the 2×2 footprint, without being given an explicit rule for the operation. In this sense, addition is not introduced as a rule but emerges as the unique operation compatible with the constraint.

Here, the height of an object is defined simply as the number of 2×2 LEGO blocks stacked vertically, so a single block has height one. When an object consisting of two black blocks is combined with one consisting of three black blocks, the result is not the number five, but rather a new object consisting of five black blocks. The numerical height provides a convenient description, but closure requires that the result be expressed as an object of the same kind as the inputs.

When objects of heights two, three, and one are combined, the final configuration depends only on the total number of blocks and is visually indistinguishable regardless of the order or grouping of combination, as shown in Figure 2. In this way, commutativity and associativity are experienced as properties of combining objects within the set, rather than as properties of the numerical heights used to describe them. The empty configuration functions as an additive identity, while removing blocks provides an intuitive though informal analogue of additive inverses. Scalar multiplication is interpreted as repeated combination, reinforcing the idea of scaling without introducing numerical formalism.

This activity is intended to build intuition rather than to serve as a formal demonstration that the collection constitutes a vector space. Because the physical constraints already enforce the axioms, students rarely question why the operation behaves as it does in this setting. Instead, the activity establishes a concrete reference point that can later be contrasted with situations in which the axioms are not automatically satisfied and must be imposed explicitly.

3.2 Activity 2: Multiple Basis Elements

In the second activity, students are again given 2×2 LEGO blocks, but now in multiple colors of black, blue, and white. The definition of the set of objects

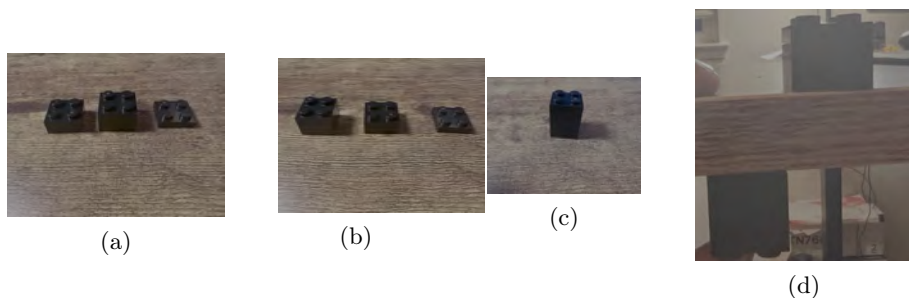


Figure 2: Commutativity, associativity, and scalar multiplication in the single-color activity. In (a), three objects of heights two, three, and one are shown. In (b), the same objects are arranged in a different order. Combining the objects in any order or grouping yields the same final configuration, shown in (c), allowing students to observe order- and grouping-independence of the operation. In (d), scalar multiplication is visualized: combining four copies of a height-one object produces a configuration above the table, while combining four copies with reversed orientation produces a configuration below the table, providing an intuitive model of positive and negative scalar multiples.

remains unchanged: admissible objects are configurations whose top-down view is a 2×2 square, with height allowed to vary. What changes is the information required to describe an object. A configuration is no longer characterized solely by its height, but by the numbers of blocks of each color it contains.

As in the first activity, students are asked to build configurations and combine them by stacking while preserving the footprint constraint. It quickly becomes apparent that any object they construct consists of some number of black blocks, some number of blue blocks, and some number of white blocks. Students naturally describe their constructions using this shared vocabulary. From this observation, it follows that these colored block types collectively generate all possible objects that can be constructed. In linear algebra terms, the set of colored blocks spans the space of admissible configurations.

To move from spanning to a basis, students are asked a deliberately simple but revealing question: whether one block type can be constructed using combinations of the others. For example, they are invited to attempt to build a white block using only blue and black blocks. Students correctly mention this is impossible, not due to lack of creativity, but because the physical constraints make such a construction impossible.

This failure becomes a pedagogical turning point. Students articulate that each block color contributes something the others cannot replace. The instructor then makes this observation explicit: if one block type could be constructed

from the others, it would be redundant and unnecessary. Because this is not possible, each color represents an independent building block. At this point, the term *linear independence* is introduced to name a constraint students have already encountered through physical reasoning.

To emphasize that a basis should also be minimal, students are asked to consider what happens if one block type is removed. For instance, if white blocks are no longer available, certain configurations become impossible to construct; the same is true if blue or black blocks are removed. From these observations, students conclude that each block type is essential: removing any one of them strictly reduces what can be built.

This discussion also provides a natural opportunity to address a common misconception raised in the single-color activity: why not treat a larger object (e.g., a configuration of four black blocks) together with a single black block as a “basis”? Students observe that the larger object can itself be built from copies of the single block, it is a scalar multiple of it, and therefore the two are linearly dependent. Including a linearly dependent element creates redundancy: the set still spans the same configurations, but it is no longer minimal. This ties linear independence directly to decomposition: an element that can be expressed in terms of the others contributes no new direction and must be excluded from a basis. Removing the four-block configuration leaves the single black block alone; a single nonzero element is linearly independent by definition, and since it generates all single-color towers by scaling, it is also a minimal spanning set and therefore a basis.

With the structure of the set and a candidate basis established, students are then asked to define an operation for combining multicolor configurations. As in the single-color case, closure continues to hold at the level of physical construction: stacking two admissible objects produces another admissible object. However, a new tension quickly emerges. When students compare configurations such as one formed by stacking four blue blocks followed by two white blocks and another formed by stacking two white blocks followed by four blue blocks, they observe that the resulting objects are physically distinct, even though addition is expected to be commutative.

This mismatch between physical appearance and algebraic expectation becomes a productive source of discussion. Some students attempt to carry over the descriptive shortcut from the single-color activity by summarizing a result with a single scalar (the total number of blocks). The instructor aligns this with the earlier case by emphasizing that, even there, the result was not a number but an object of the same kind (e.g., “nine black blocks” rather than “height nine”). Students then ask whether an analogous single-scalar description can serve as the result in the multicolor setting. They recognize that it cannot: a single total discards the color information that is part of what defines the

objects in this collection.

This moment sharpens the meaning of closure and representation simultaneously. Addition must take two objects from the collection and produce another object of the same type, expressed using the same information used to describe the elements of the set. In the multicolor setting, being an element of the set requires retaining the counts of each color, not collapsing them into a single total.

To resolve the conflict between representation and commutativity, students are asked what features the two configurations have in common. Through discussion, they identify that the numbers of blocks of each color are the same regardless of order. This observation motivates a refinement of representation. Rather than viewing an object as a physical stacking arrangement, it is described by a triple (b, u, w) recording the number of black, blue, and white blocks it contains. Once students accept that a configuration such as five white blocks, two blue blocks, and one black block is an element of the set, a natural question arises: Since the individual blocks are themselves elements of the same set, they must also be describable using the same basis used to describe all other objects in the collection. This leads to an important identification. A single black block is represented by $(1, 0, 0)$, a single blue block by $(0, 1, 0)$, and a single white block by $(0, 0, 1)$. In this way, the colored building blocks are not merely generators of the space but are explicitly identified with basis vectors in the coordinate representation. More complex configurations are then understood as linear combinations of these basis vectors, with coefficients given by the corresponding block counts.

With this representation in place, addition is no longer ambiguous. Combining two objects corresponds to adding their coordinates component-wise,

$$(b_1, u_1, w_1) + (b_2, u_2, w_2) = (b_1 + b_2, u_1 + u_2, w_1 + w_2),$$

producing a result that remains within the same set and is independent of the order of combination. In this way, coordinate-wise vector addition is not introduced as an arbitrary rule, but emerges as a necessary abstraction forced by the requirement that results remain expressible in the same terms as the objects being combined.

At this point, students are effectively working in $\mathbb{R}^3$: they are adding and scaling vectors whose coordinates record contributions from each independent basis element. The LEGO activity makes this transition tangible, allowing students to see that once a collection of objects is recognized as a vector space with a chosen basis, the same operations and structural reasoning apply regardless of whether the objects are blocks, coordinate triples, polynomials, or functions.

As in the single-color activity, the additive identity corresponds to the

empty configuration. Additive inverses cannot be represented directly as negative block counts; instead, they are introduced through cancellation, using the signed-count convention, where blocks above a reference level (such as a table top) are assigned positive count and blocks below are assigned negative count. The inverse of a configuration is the object that, when combined with it, returns the empty configuration $(0, 0, 0)$; physically, this means balancing each color count to zero, either by removing blocks or by adding their opposites across the reference level. To support visualization, a reference plane is introduced: configurations above it are interpreted as having positive count, while an equivalent configuration designated as below it is assigned a negative count by convention. The reference plane itself, corresponding to zero blocks, represents the empty configuration $(0, 0, 0)$. Within this convention, negative scalar multiplication is introduced conceptually rather than physically. Multiplying by -1 is described as producing a configuration that cancels the original, reinforcing the idea that inverses operate component-wise.

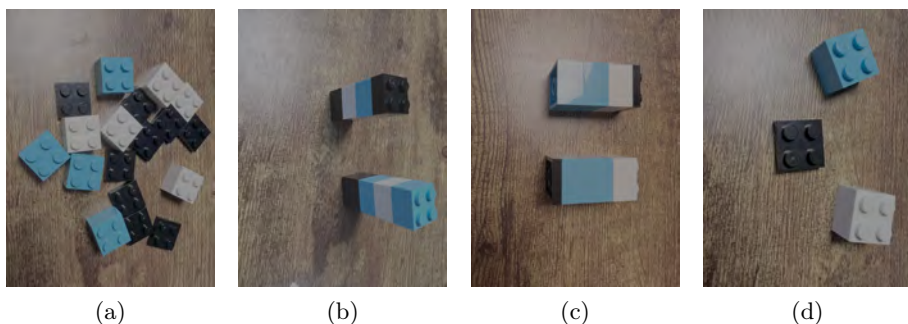


Figure 3: Multi-color activity depicting basis elements and representation. In (a), students are given an assortment of black, blue, and white 2×2 LEGO blocks. In (b), students construct a valid configuration that belongs to the set, defined by having a 2×2 footprint when viewed from above, using a particular ordering of colored blocks. In (c), combining the same colored blocks in a different order yields a physically distinct configuration, revealing a tension with commutativity despite closure holding at the level of construction. In (d), this tension is resolved by representing configurations in terms of their color counts: all admissible objects, including those in (b) and (c), are understood as linear combinations of the same three basis elements corresponding to the black, blue, and white blocks.

4 Discussion and Reflections

This activity highlights a distinction that is often implicit in traditional instruction: the difference between algebraic structure and physical representation. In the single-color case, the physical model already satisfies the axioms, and abstraction is minimal. In the multi-color case, abstraction becomes necessary to reconcile physical appearance with algebraic requirements. Students do not merely verify axioms; they experience why certain definitions are forced. LEGO serves as an effective medium for this exploration because it supports experimentation, makes failures visible, and encourages discussion.

4.1 Anticipating Questions of Rigor

A likely concern is whether the use of LEGO oversimplifies the mathematics or obscures formal rigor. We emphasize that the activity is not intended as a formal proof of the vector space axioms, but as a preparatory experience that supports later formalization. The transition from physical configurations to coordinate representations is made explicit throughout: students are consistently guided to recognize that the LEGO constructions are concrete representations that motivate the abstract structure, not the mathematical objects themselves.

A related concern involves the finiteness of the physical materials. In practice, students work with a finite collection of LEGO blocks, and once a block is used in a particular construction it is temporarily unavailable for others. This limitation is addressed explicitly in discussion. Students are reminded that the mathematical setting being modeled is abstract and does not depend on the availability of physical pieces: a block used in one configuration continues to exist conceptually and may be reused in other combinations. When necessary, additional blocks are introduced during the activity to reinforce this point.

A further limitation concerns scalar multiplication. In the LEGO setting, scaling an object corresponds to repeated combination or cancellation, and is therefore restricted to integer scalars, both positive and negative. Fractional scalars, such as $\frac{1}{2}$, have no direct physical interpretation in this context: there is no meaningful way to construct half of a block. This restriction is made explicit to students. The instructor emphasizes that the formal definition of a vector space requires scalars to belong to a field, typically $\mathbb{R}$, and that the integers $\mathbb{Z}$ do not form a field since they lack multiplicative inverses: for example, $\frac{1}{2} \notin \mathbb{Z}$. The LEGO activity therefore serves as an intuitive entry point, with the understanding that scalar multiplication over $\mathbb{R}$ is introduced later through formal definitions and symbolic representations.

By making these distinctions explicit, students learn to separate the constraints of the physical model from the properties of the underlying mathematical structure. The 2×2 LEGO blocks is not treated as a perfect copy of a

vector space, but as a concrete setting for reasoning about selected structural ideas, such as allowable objects, closure, and operations, before those ideas are formalized symbolically.

4.2 Instructor Observations and Reflections

Across multiple implementations of this activity, a recurring and instructive pattern emerged in how students reasoned about addition. In the single-color case, several students naturally interpreted addition as total height, an interpretation that worked seamlessly: stacking three blocks of black on two blocks of black produced a tower of height five black blocks, and the result was physically identical regardless of order. In this setting, height alone fully described the object, and students experienced no tension between the physical model and the algebraic properties of addition.

This intuition carried over when multiple block colors were introduced. At this stage, the individual colors began to function as basis blocks: each tower could be described by recording how many blocks of each color it contained. For example, using the coordinate order black, blue, and white, a tower with four black blocks and two blue blocks could be represented as $(4, 2, 0)$. Some students nevertheless proposed defining the sum of such a tower as a single number representing total height. While this proposal was mathematically consistent with their prior experience, discussion quickly revealed its limitations. Treating the result as a single number ignored the basis-block description: different colored towers with the same total height, such as $(4, 2, 0)$ and $(3, 3, 0)$, became indistinguishable. In effect, this move reduced a situation involving multiple independent components to a one-dimensional description, returning to the earlier single-color case. Importantly, the problem was not that the total height was wrong, but that it produced a scalar rather than another colored tower described in terms of the same basis blocks.

The exploratory nature of the LEGO activity played a central role in facilitating this insight. Students were encouraged to combine blocks freely, including placing them side by side or offsetting them, and to evaluate the results against the agreed-upon definition of a valid object. These explorations made failures of closure immediately visible: side-by-side combinations produced shapes that no longer had a 2×2 footprint when viewed from above. Through experimentation and discussion, students converged on vertical stacking as the only operation that preserved the defining features of the objects under consideration. From an instructional perspective, this exploration reframed vector addition as a deliberately chosen operation rather than a given rule. Students came to see that addition must be defined in a way that respects the structure of the objects and satisfies the axioms of a vector space. The axioms (we list in Table 1 all the axioms of vector space and the mapping

across the two activities) were then used not merely to verify correctness, but to justify why certain definitions were acceptable and others were not.

5 Conclusion

By engaging students in the design of addition under explicit constraints, the LEGO activity makes the vector space axioms tangible and meaningful. The contrast between single-color and multi-color cases reveals when algebraic properties align naturally with physical structure and when abstraction is required. As a reflective teaching experience, the activity provides students with a memorable foundation for understanding vector spaces as structured objects governed by carefully chosen operations.

Acknowledgment

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

References

- Aytekin, C., & Kiymaz, Y. (2019). Teaching linear algebra supported by geogebra visualization environment. *Acta Didactica Napocensia*, 12(2), 75–96.
- Beltrán Meneu, M. J., & Murillo Arcila, M. (2016). Coloreando el álgebra lineal. *Modelling in Science Education and Learning*, 9(1), 25–34.
- Berman, A., & Shvartsman, L. (2016). Definitions are important: The case of linear algebra. *European Journal of Science and Mathematics Education*, 4(1), 26–32.
- Cook, E. (2022). Stop-motion lego® animations for learning linear algebra. *International journal of mathematical education in science and technology*, 53(3), 594–602.
- Cuevas-Vallejo, A., Orozco-Santiago, J., & Paz-Rodríguez, S. (2023). A learning trajectory for university students regarding the concept of vector. *The Journal of Mathematical Behavior*, 70, 101044.
- Dereziński, M., & Mahoney, M. W. (2024). Recent and upcoming developments in randomized numerical linear algebra for machine learning. *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 6470–6479.
- Dorier, J.-L. (2003). Teaching linear algebra at university. *arXiv preprint math/030501*

- Dorier, J.-L., Robert, A., Robinet, J., & Rogalski, M. (2000). On a research programme concerning the teaching and learning of linear algebra in the first-year of a french science university. *International Journal of Mathematical Education in Science and Technology*, 31(1), 27–35.
- Dorier, J.-L., & Sierpinska, A. (2001). Research into the teaching and learning of linear algebra. In D. Holton, M. Artigue, U. Kirchgräber, J. Hillel, M. Niss, & A. Schoenfeld (Eds.), *The teaching and learning of mathematics at university level: An icmi study* (pp. 255–273). Springer Netherlands. https://doi.org/10.1007/0-306-47231-7_24
- Fernández-Cézar, R., Herrero, H., Pla, F., & Solares, C. (2020). Is there any impact of teaching vector spaces from real problems? the case of first year engineering students. *Journal of Research in Science, Mathematics and Technology Education*, 3(3), 125–139. <https://doi.org/10.31756/jrsnte.332>
- Kannan, R., & Vempala, S. (2017). Randomized algorithms in numerical linear algebra. *Acta Numerica*, 26, 95–135.
- Lessing, M.-J., & Ogbonnaya, U. I. (2025). Reasoning and proof in mathematics education: A systematic literature review. *Discover Education*.
- Stewart, S., & Thomas, M. O. (2019). Student perspectives on proof in linear algebra. *ZDM*, 51(7), 1069–1082.
- Trigueros, M., Cabrera, A. C., & Aguilar, M. S. (2024). Mental constructions for the learning of the concept of vector space. *ZDM–Mathematics Education*, 56(7), 1417–1431.
- Turgut, M., Smith, J. L., & Andrews-Larson, C. (2022). Symbolizing lines and planes as linear combinations in a dynamic geometry environment. *The Journal of Mathematical Behavior*, 66, 100948.

6 Appendix

Table 1: Activity 1 uses vertical stacking for addition and a formal cancellation convention for opposites. Activity 2 resolves a tension between physical arrangement and algebraic expectations by adopting a signed color-count representation (b, u, w) . Physical scaling models only integer multiples; real scalar multiplication and the full vector space structure are realized only after passing to the abstract coordinate model $\mathbb{R}^3$ that the activities motivate.

Axiom / Property	Activity 1: Single color (black)	Activity 2: Multi-color (black/blue/white)
Closure under addition	Students are told the set consists of configurations with a 2×2 footprint. They test combinations and discover that vertical stacking is the only combination that stays in the set.	Physical stacking preserves the 2×2 footprint, but students see that the result must also be representable in the same form as the inputs: as a color-count triple (b, u, w) , not merely as a total height.
Commutativity ($x + y = y + x$)	Two objects combined in either order produce an indistinguishable final configuration; order independence is visually apparent.	With physical order as the representation, swapping order can change the visible arrangement. This motivates redefining the object by color counts (b, u, w) so that order does not matter and commutativity is restored component-wise.
Associativity ($(x + y) + z = x + (y + z)$)	Students combine three objects, such as towers of heights 2, 3, and 1, using different groupings and observe the same final configuration; grouping independence is physically visible.	After adopting the (b, u, w) representation, grouping corresponds to component-wise addition, so different groupings yield the same triple. Before the representation shift, grouping can change intermediate physical arrangements, creating productive tension.
Additive identity ($x + 0 = x$)	The empty configuration, consisting of no blocks, leaves any object unchanged when combined with it. Students can interpret this as “doing nothing.”	The same identity element appears in coordinates as $(0, 0, 0)$. It is interpreted as the empty configuration; adding it leaves all color counts unchanged.
Additive inverses ($x + (-x) = 0$)	Physical towers do not contain literal negative block counts. Additive inverses therefore require extending the physical stacking model to a formal cancellation convention: each positive tower is paired with an opposite that undoes or cancels it, returning to the empty configuration.	Inverses are introduced component-wise through formal signed color counts: $(b, u, w) + (-b, -u, -w) = (0, 0, 0).$ Physically, students reason in terms of canceling or removing blocks by color; mathematically, this extends nonnegative color counts to signed counts.
Closure under scalar multiplication αx	Scalar multiplication begins as repeated combination or cancellation: $\alpha \cdot x$ means stacking α copies above the reference plane for $\alpha \in \mathbb{Z}^+$, and $ \alpha $ copies below it for $\alpha \in \mathbb{Z}^-$, with $0 \cdot x$ returning the empty configuration. This physically models integer scaling only. Real scalars, such as $\frac{1}{2}$, require an abstract extension beyond literal block counts, since the integers $\mathbb{Z}$ do not form a field and fractional blocks have no physical meaning in this context.	After the coordinate shift, integer scaling is represented component-wise: $\alpha(b, u, w) = (\alpha b, \alpha u, \alpha w), \quad \alpha \in \mathbb{Z},$ where positive values correspond to configurations above the reference plane and negative values correspond to configurations below it. The formal definition of a vector space requires scalars to belong to a field, so the activity uses LEGO blocks to motivate the structure and then extends it abstractly to $\alpha(b, u, w) = (\alpha b, \alpha u, \alpha w), \quad \alpha \in \mathbb{R}.$
Distributivity over vector addition $\alpha(x + y) = \alpha x + \alpha y$	For integer scalars, students can compare “scale then combine” with “combine then scale” by repeating stacks above or below the reference plane; the outcomes match in total signed count color by color. For real scalars, this property belongs to the abstract extension of the model, since fractional block counts have no physical meaning and real scalars require the full field structure of $\mathbb{R}$.	For integer scalars, distributivity is visible through repeated color counts. In the abstract coordinate model, it becomes $\alpha((b, u, w) + (b', u', w')) = \alpha(b, u, w) + \alpha(b', u', w'),$ which holds component-wise for real scalars.
Distributivity over scalar addition ($(\alpha + \beta)x = \alpha x + \beta x$)	For integer scalars, combining α copies and β copies of the same object yields $(\alpha + \beta)$ copies of that object. For real scalars, this is treated as part of the abstract algebraic extension.	In coordinates, $(\alpha + \beta)(b, u, w) = \alpha(b, u, w) + \beta(b, u, w) = (\alpha b + \beta b, \alpha u + \beta u, \alpha w + \beta w)$ aligning with repeated combination for integer scalars and extending algebraically to real scalars.
Compatibility of scalar multiplication ($(\alpha\beta)x = \alpha(\beta x)$)	For integer scalars, repeated scaling is associative: making β copies of an object and then α copies of that result gives the same configuration as making $\alpha\beta$ copies at once.	In coordinates, scalar compatibility holds component-wise: $(\alpha\beta)(b, u, w) = \alpha(\beta(b, u, w)).$ For LEGO blocks this is physically meaningful for integer scalars; for real scalars it is part of the abstract coordinate model.
Scalar identity $1x = x$	One copy of any object is the object itself: $1 \cdot x = x$.	Basis-block identification becomes explicit: black $\mapsto (1, 0, 0)$, blue $\mapsto (0, 1, 0)$, white $\mapsto (0, 0, 1)$, so $1 \cdot (b, u, w) = (b, u, w)$.

Discussions of NP-Complete Problems: An Academic Website*

Sean McCulloch
Ohio Wesleyan University
Delaware, OH 43015
`stmccull@owu.edu`

Abstract

The study of NP-complete problems is an important topic covered in many Analysis of Algorithms and Theory of Computation courses. In order to present a wider range of problems to use in these classes, we have created a website that contains an article for each problem in the appendix of Garey & Johnson's classic Computers and Intractability book, as well as several problems not in the appendix. Each article describes an example of the problem under discussion, a sketch of a reduction showing the problem as NP-hard, and a subjective difficulty rating of the problem. In building this website, we identified several problems with no cited reduction in Garey & Johnson, and our website provides the reductions for these problems as well—providing an explicit source for problems whose NP-completeness has long been cited without an accompanying proof.

1 Introduction

The task of finding the run-time complexity of an algorithm is a central one to Computer Science. An important aspect of the complexity of a problem is whether it is NP-complete. Beginning with seminal papers from Cook [1] and Karp [2], hundreds, if not thousands, of problems have been shown to be NP-complete, which gives strong evidence that no polynomial-time algorithm to solve the problem will be found. Problems that are NP-complete (or harder)

*Copyright ©2026 by the Consortium for Computing Sciences in Colleges. Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the CCSC copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Consortium for Computing Sciences in Colleges. To copy otherwise, or to republish, requires a fee and/or specific permission.

have been found in many areas across the breadth of Computer Science, including Artificial Intelligence [3][4], Compilers [5][6], VLSI design [7][8], and Databases [9][10]. The study of NP-completeness is seen as a vital part of a student's undergraduate education, usually in an Analysis of Algorithms or Theory of Computation class, as evidenced by its inclusion as a "core topic" in the ACM 2023 curriculum guidelines [11]. As such, most major textbooks used in Algorithms classes (e.g.: [12][13][14][15]) devote significant space to the discussion of NP-completeness. However, in these books, most of the treatment is done through the same example problems, mainly from the original Karp paper: problems such as Satisfiability, Vertex Cover, Clique, and Subset Sum. An instructor may naturally wish to have a wider range of problems to use as in-class examples or for homework assignments or exam questions.

Originally published in 1979, Michael Garey and David Johnson's Computers and Intractability: A Guide to the Theory of NP-completeness[16] is a core work in the field of NP-completeness, and even today is seen as the main reference work explaining the topic. In it, Garey and Johnson discuss what it means for a problem to be NP-complete, ways to prove a problem as NP-complete, and ways to cope with problems that are NP-complete. It also contains an appendix with brief descriptions of several hundred NP-complete problems that were known at the time. This appendix is potentially a treasure trove of ideas for the educator, but the problem descriptions are too brief to be easily used in a classroom.

To help remedy this, we have developed an academic website [17] that contains an article for each problem in the appendix, as well as several additional problems that were deemed useful. For each problem, the article provides:

- The name and abbreviation of the problem. The problems in Garey and Johnson's book are given abbreviations. For example, Vertex Cover is abbreviated "GT1" since it is the first problem listed in the Graph Theory section of the appendix.
- A description of the problem, which, where needed, expands upon the basic description from the appendix to give more detail and add clarification.
- An example of the problem. Since all problems are phrased as decision problems, we attempt to provide a "yes" and "no" example for most problems.
- An overview of the reduction proving the problem is NP-hard. Where possible, the reference to the actual publication with the reduction is provided and a summary of the reduction is provided in the article. For other problems where the publication is not available, we attempt to provide our own reduction to the problem.
- A subjective difficulty rating of the problem, from the point of view of an

instructor looking to assign the reduction to a class of students learning about NP-completeness. The ratings range from 1 (trivial) through 10 (virtually impossible).

The website also provides a bibliography of nearly 200 references to publications discussing the various problems seen on the site. To protect problems from students looking for homework answers, most articles are hidden behind a password that is available by contacting the author.

2 A Brief Review of NP-completeness

The study of NP-complete problems is an important one when learning how to design algorithms, partially because the “Does $P=NP$?” question is perhaps the most important open question in Computer Science. Showing a problem to be NP-complete is strong evidence that no polynomial-time algorithm to solve the problem exists, freeing one to look for alternative solutions to the problem. Additionally, all problems that are NP-complete are related in the sense that a polynomial-time solution to any one NP-complete problem (or a proof that no polynomial-time algorithm can possibly exist for that problem) will also give a polynomial-time solution (or a proof that no polynomial-time algorithm exists) for all NP-complete problems.

The problems under consideration for this topic are decision problems—problems whose specific instances always return “true” or “false”, analogous to Boolean functions. Optimization problems can be turned into a decision problem by adding an extra parameter.

Showing a problem is NP-complete involves two parts. The first is to show that the problem belongs in class “NP”, which is the class of problems that can be solved non-deterministically in polynomial time. There are several ways to do this: in our classes, we follow the model given in Cormen[12], where we are given an instance of a problem, and a possible solution to the problem, and need to verify that the possible solution is correct in polynomial time.

The more difficult part in showing a problem is NP-complete is showing it is NP-hard. The method by which we show this is by using a *polynomial-time reduction*, where we show that if our problem (which we will call *Prob*) can be solved in polynomial time, then the known NP-complete problem (which we will call *KnownNP*) can be solved in polynomial time as well. We explain this concept to students in terms of function calls: We write a function for a known NP-complete problem that calls our problem as a subfunction. Our function takes an instance *I* of *KnownNP*, builds an instance *I'* of *Prob*, and calls the solver.

Listing 1: General form of a polynomial-time reduction as a function

```
bool KnownNP_Solver(KnownNP_Instance I){  
    I' := Convert_Instance(I);
```

```

if (Prob_Solver(I'))
    return true;
else
    return false;
}

```

We then need to show:

- The conversion is in polynomial time.
- The *KnownNP* solver returns true on I if and only if our *Prob* solver returns true on I'

The reduction section of our articles focuses on describing the conversion from I to I' , and a sketch of the $yes_{Prob} \iff yes_{KnownNP}$.

3 Article Structure

Our website has one article for each of the problems in Garey & Johnson's appendix. The first few articles discuss the "basic" NP-complete problems used in Chapter 3 of the book, and used by Garey & Johnson as examples for many reductions: 3-Satisfiability (3SAT), 3-Dimensional Matching (3DM), Vertex Cover (VC), Clique, Hamiltonian Circuit (HC), and Partition.

We then go through the appendix from start to finish. Along the way, we also provide some articles on other problems that do not appear in the appendix. Some of these are problems used in reductions for appendix problems, others are useful extensions of appendix problems. In one case, a student doing an independent study wrote an article showing a generalization of the video game "Pokemon" was NP-hard, reducing from a problem in the appendix.

Each article has several sections. An article begins with the name of the problem, including the abbreviated name used in the appendix of Garey & Johnson. Then we describe the problem. Oftentimes, the description in Garey & Johnson's appendix benefits from more explanation, and that explanation is provided. Figure 1 shows these sections for a problem called "Clustering" that is perhaps less well-known than other NP-complete problems, but is still approachable by students learning this topic. In this problem, we have changed the framing of the problem slightly from how it was presented in the appendix: there the instance is defined by a set X of items, with a defined positive distance between each pair. Representing this structure as a graph makes the problem easier to understand, so that is how the definition is given.

The next section of the article is an example. Each article provides an example of the problem, so the reader can see the various terms and variables in use to better grasp the meaning of the problem. Where possible, we try to show an example that can, with a small change, be converted between a "yes" instance and a "no" instance. Showing both kinds of instances gives intuition about how the problem works, and we have found that showing instances that are similar, but give different results helps give intuition about how NP-complete problems

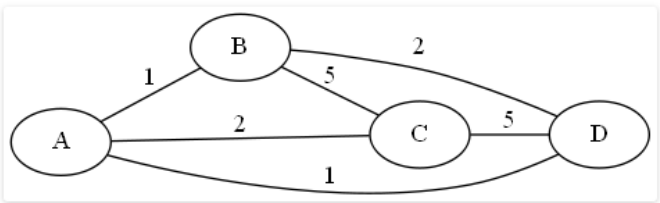
The problem: Clustering. This is problem MS9 in the appendix.

The description: Given a complete weighted graph $G=(V,E)$, with a positive weight $d(i,j)$ for each edge e_{ij} , and two positive integers K and B . Can we partition V into K (or less) disjoint sets $V_1..V_k$ such that within each V_i , all edges that stay within the partition cost B or less?

Figure 1: Name and description from Clustering article.

often have subtle factors that make them hard to solve. Figure 2 shows the example given in the Clustering article.

Example: Here is a graph:



If $K=2$, and $B=2$, we could have the sets $\{A,C\}$ and $\{B,D\}$ as a legal cluster. But if $K=2$ and $B=1$, we will not be able to solve the problem. For example, we might want to have the edges (A,B) and (A,D) in the same cluster, but that would require us also to consider the edge (B,D) , which is larger than 1.

Figure 2: Example from Clustering article.

Often the largest part of the article is the section discussing the reduction showing the problem is NP-hard. If the reduction is simple and straightforward (as is the case for Clustering, as Figure 3 shows), then just showing how the conversion happens is sufficient to explain the reduction. For more difficult problems, this section is longer, and describes how we would arrive at both directions of the equivalence proof. Figure 4 shows an example of a longer reduction from the “Integer Knapsack” article. Integer Knapsack is like 0/1 Knapsack, but we are allowed to take an item multiple times.

The article concludes with a subjective rating of the difficulty of the problem’s reduction, from the point of view of a student learning about NP-completeness. Generally speaking, ratings of 1-2 indicate that the conversion is trivial- often a restatement of the known NP-complete problem. Ratings of 3-6 are problems that need some significant conversion in the reduction, but that conversion can be understood by students. Ratings of 7 or higher often are too long, too complex, or require very complicated “widgets” to complete the reduction. Some of them might be good for a classroom discussion, but we

feel that they are too difficult to expect a student to figure out on their own. Thus, we recommend using problems in the 3-6 rating range as homework and exam problems for an undergraduate course, tending towards the lower end of that range if (as has been our experience), this topic only sees a week or two of class time during the course.

Reduction: [Brucker's paper](#) introduces the problem using a graph, like I did above. (G&J talk about points and distances), His reduction is from [Graph Coloring](#).

Suppose we have a coloring graph G . We build a new graph G' such that the weight of an edge in G' is 1 if it did *not* appear in G , and its weight is 2 if it *did* appear in G . Then we set our K = the K of the coloring problem (3 for 3-coloring), and $B = 2$. A partition of the vertices into 3 sets where each set has no weight-2 edges is exactly a legal coloring.

Difficulty: 3. I do think the graph terminology helps to make the problem more understandable.

This entry was posted in [Appendix: Miscellaneous](#) and tagged [Clustering](#), [Difficulty 3](#), [Graph 3-coloring](#), [MS9](#) on [September 29, 2023](#). [Edit](#)

Figure 3: Reduction, difficulty rating, and tags from Clustering article.

The article concludes with a set of tags. Each article has a tag for its rating, for itself, and for the article containing the problem the reduction uses to reduce from. These links help the site show the connections between all of the problems on the site.

While we wish to make these articles available to any educator or researcher who wants to use them, we also want to limit their use by students hunting for homework solutions, or by data mining tools scraping the internet for information. Thus, we have instituted a policy of making the most recent 10 articles publicly readable. The remaining articles are protected by a password that can be obtained by contacting the site's author.

Reduction: It really seems very close to regular Knapsack, doesn't it? G&J even say that the reduction is from Sum of Subsets, just like our Knapsack reduction did. But the [reference from Lueker](#) is actually pretty complicated:

We're given a Sum of Subsets instance- a set S with n elements and a value K to try to hit exactly.

Define M to be $1 + \max(s_i)$, and 2^n . We'll now create a set U for our Integer Knapsack instance with $2n$ elements:

Each element will be labeled u_{ij} , where i runs from 1 to n and j runs from 0 to 1. Element u_{ij} will have profit and weight $n \cdot M^{n+1} + M^i + j \cdot s_i$

Our K' (and B') will be: $n \cdot M^{n+1} + \sum_{i=1}^n M^i + K$.

If S has a subset that sums to K , then for each element i , if we took s_i in the solution, we take u_{i1} in our Integer Knapsack solution. Otherwise, we take u_{i0} .

If U has an Integer Knapsack solution, we know that we have non-negative integer values c_{ij} such that $\sum_{i=1}^n \sum_{j=0}^1 c_{ij} \cdot u_{ij} = K'$. Because of the way we chose M , if we divide K' by M^{n+1} and round down, we get n . Because each u_{ij} has M^{n+1} in it, dividing the summation side by M^{n+1} gets us a number at least as big as the sum of the c_{ij} values. So we can say that $\sum_{i=1}^n \sum_{j=0}^1 c_{ij} \leq n$. Thus, each element is taken at most n times.

Now we can go back to the original equation: $\sum_{i=1}^n \sum_{j=0}^1 c_{ij} \cdot u_{ij} = K'$ and do algebra. We learn that in fact, each c_{ij} is 0 or 1, and that $c_{i0} + c_{i1} = 1$ for all i (so we never take both c_{i0} and c_{i1}), and that the values of c_{i1} are 0/1 solutions to the corresponding elements in the Sum of Subsets problem.

Figure 4: Reduction section from Integer Knapsack article

4 Uncited Reductions

The appendix of Garey & Johnson's book lists 320 problems, each of which has an article described on the website. Most of the descriptions in the appendix just describe the problem to reduce from, without giving the details of the reduction. To learn the details of many of the reductions, we consulted the paper publishing the result, placing a reference in our site's bibliography. Thus, the site's bibliography contains over 170 references to papers (mostly from the 1970s) containing the reductions to many of these problems.

However, many of the problems presented in the appendix do not have an easily accessible reference. Many of the citations in the book are to "private communications" that are not publicly available, or to "unpublished manuscripts" that may or may not have been subsequently published. In some cases, we were able to find and use more recent citations— some by the authors

referenced in the appendix, some not— but in others, we could not find a reference to use. There are approximately 25 such problems, and in these cases, we have attempted to create and describe the reduction to the problem ourselves.

In practice, authors often cite a problem listed in the appendix as evidence of its NP-hardness. This occurs even in cases where no easily traceable reference or accompanying reduction is provided there. The reductions presented in our articles help address this gap.

5 Some Useful Less Popular NP-complete Problems

Our initial goal in creating this site was to investigate other problems that are good examples to use in courses to explain NP-completeness to students beyond the “standard” ones such as Satisfiability, Vertex Cover, 0/1 Knapsack, or Traveling Salesman. As such, we present some brief descriptions of problems that an educator may find interesting. Our ratings for these problems are all in the 2-4 range, making them hopefully very approachable by a student working alone on a homework assignment or exam:

- **Clustering** (Described in our example in section 3). Given a weighted, undirected graph, and integers B and K , can we partition the vertices of the graph into at most K sets, such that the cost of each set’s subgraph totals B or less? (Reduction from Graph Coloring)
- **Minimum Sum of Squares** Given a set A of integer elements, and integers K and J , can we partition A into at most K sets A_i such that $\sum_i (\sum_{a \in A_i} (a^2)) \leq K$? (Reduction from Partition)
- **Hitting String** Given a set of n -length strings A over the alphabet $0,1,*$, is there an n -length string s over the alphabet $0,1$, where s agrees in at least one position with each element of A ? (Reduction from 3SAT)
- **Set Packing** Given a collection C of finite sets, and a positive integer K . Does C contain K disjoint sets? (Reduction from Exact Cover on 3-Sets)
- **Biconnectivity Augmentation** Given a graph G with a weight between each pair of vertices (even ones that do not have an edge), and a positive integer B , does there exist a set E' of edges we can add to G , of cost at most B , that will make our graph be biconnected- still connected after removing any one vertex? (Reduction from Hamiltonian Cycle)

There are also many versions of Satisfiability problems, or of Scheduling problems, that fit into this range as well.

6 Conclusion

Garey & Johnson’s book has described several hundred NP-complete problems, many of which are suitable examples for use in an undergraduate classroom. We have created this website to be accessible to instructors looking for a wider breadth of problem examples. The expanded problem descriptions, examples, and reduction sketches are all written to this audience, as opposed to a re-

searcher in the field.

In the process of constructing these articles, we identified a number of problems whose NP-hardness is widely accepted but not supported by an easily accessible published reduction. By documenting reductions for these cases, the site also serves as a form of verification. Now that coverage of the appendix is complete, the site can function as a stable reference resource for instructors teaching NP-Completeness, with the possibility of future expansion to related algorithmic topics.

References

- [1] Stephen A. Cook. “The complexity of theorem-proving procedures”. In: *Proceedings of the Third Annual ACM Symposium on Theory of Computing*. STOC ’71. Shaker Heights, Ohio, USA: Association for Computing Machinery, 1971, pp. 151–158.
- [2] Richard M. Karp. “Reducibility among Combinatorial Problems”. In: *Complexity of Computer Computations: Proceedings of a symposium on the Complexity of Computer Computations*. Ed. by Raymond E. Miller, James W. Thatcher, and Jean D. Bohlinger. Boston, MA: Springer US, 1972, pp. 85–103.
- [3] Tom Bylander. “Complexity results for planning”. In: *Proceedings of the 12th International Joint Conference on Artificial Intelligence - Volume 1*. IJCAI’91. Sydney, New South Wales, Australia: Morgan Kaufmann Publishers Inc., 1991, pp. 274–279.
- [4] Mark Huasong Meng et al. “Adversarial Robustness of Deep Neural Networks: A Survey from a Formal Verification Perspective”. In: *IEEE Transactions on Dependable and Secure Computing* (2022), pp. 1–1. DOI: 10.1109/TDSC.2022.3179131.
- [5] John Bruno and Ravi Sethi. “Code Generation for a One-Register Machine”. In: *J. ACM* 23.3 (July 1976), pp. 502–510.
- [6] Gregory J Chaitin. “Register allocation & spilling via graph coloring”. In: *ACM Sigplan Notices* 17.6 (1982), pp. 98–101.
- [7] M. R. Garey and D. S. Johnson. “The Rectilinear Steiner Tree Problem is $\$NP\$$ -Complete”. In: *SIAM Journal on Applied Mathematics* 32.4 (1977), pp. 826–834.
- [8] O.H. Ibarra and S.K. Sahni. “Polynomially Complete Fault Detection Problems”. In: *IEEE Transactions on Computers* C-24.3 (1975), pp. 242–249.
- [9] Catriel Beeri and Philip A. Bernstein. “Computational problems related to the design of normal form relational schemas”. In: *ACM Trans. Database Syst.* 4.1 (Mar. 1979), pp. 30–59.
- [10] Cláudio L. Lucchesi and Sylvia L. Osborn. “Candidate keys for relations”. In: *Journal of Computer and System Sciences* 17.2 (1978), pp. 270–279.
- [11] Amruth N. Kumar et al. *Computer Science Curricula 2023*. New York, NY, USA: Association for Computing Machinery, 2024.
- [12] Thomas H. Cormen et al. *Introduction to Algorithms*. Cambridge, MA: The MIT Press, 2022.

- [13] Sanjoy Dasgupta, Christos Papadimitriou, and Umesh Vazirani. *Algorithms*. McGraw-Hill Education, 2006.
- [14] Jon Kleinberg and Eva Tardos. *Algorithm Design*. Boston, MA: Pearson Education, 2005.
- [15] Anany Levitin. *Introduction to the Design and Analysis of Algorithms*. Boston, MA: Pearson Education, 2011.
- [16] Michael R. Garey and David S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. USA: W. H. Freeman & Co., 1990.
- [17] Sean McCulloch. *Discussions of NP-Complete Problems: A Resource for Computer Science Educators*. <https://npcomplete.owu.edu/>.

An Explainable Artificial Intelligence Framework for Personalized Diabetes Risk Stratification*

Shruti Bhandari, Sanjog Dhakal, Robin Ghosh, Tolga Ensari
Division of Computing Sciences
Arkansas Tech University
Russellville, AR 72801

{sbhandari3, sdhakal3, rghosh, tensari}@atu.edu

Abstract

Type 2 diabetes mellitus is an increasing global health concern, highlighting the need for early identification of individuals at elevated risk to reduce disease progression and long-term complications. This study presents an interpretable machine learning framework for diabetes risk stratification that distinguishes between medium and high-risk populations while maintaining clinical transparency. A dataset of approximately 70,000 patient records with 34 demographic, metabolic, and lifestyle features was analyzed using Logistic Regression, Decision Tree, Random Forest, XGBoost, and LightGBM within a unified preprocessing pipeline. Class imbalance was handled using weighting strategies, and model interpretability was achieved through SHAP, LIME, and partial dependence analysis. Ensemble and boosting-based models, particularly LightGBM, demonstrated superior discriminatory performance, achieving an accuracy of 0.99 and ROC AUC of 0.999 while maintaining efficient inference times. Across all explainability methods, blood glucose levels, body mass index, waist circumference, cholesterol components, and pregnancy-related weight changes consistently emerged as the most influential risk factors. Probability-based predictions enabled individualized risk scoring and clear stratification of patients into low, medium,

*Copyright ©2026 by the Consortium for Computing Sciences in Colleges. Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the CCSC copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Consortium for Computing Sciences in Colleges. To copy otherwise, or to republish, requires a fee and/or specific permission.

and high-risk categories. These findings suggest that the proposed explainable machine learning approach provides accurate and transparent diabetes risk stratification, supporting early intervention and personalized prevention strategies.

Keywords: diabetes mellitus, risk classification, interpretable AI, machine learning, shap, lime, predictive analytics

1 Introduction

This study presents an explainable machine learning framework for personalized diabetes risk stratification that integrates probabilistic risk scoring, multi-model evaluation, and complementary explainability techniques within a single pipeline. Interpretability is treated as a core component rather than a post-processing step, allowing predictions to be examined alongside their contributing features.

A central technical contribution of this research is reformulating diabetes prediction as a probability-driven risk stratification task. Instead of relying solely on binary outcomes, predicted probabilities are used to assign individuals to low, medium, and high-risk categories. This formulation enables personalized risk assessment while preserving population-level interpretability. The same probability-based stratification strategy is applied consistently across linear tree-based ensemble and gradient boosting models, allowing direct comparison of predictive behavior under a unified risk definition.

Another important contribution is the structured integration of multiple explainability methods to analyze model behavior at different levels. Global feature relevance is quantified using mean absolute SHAP values to identify predictors that consistently influence high-risk classification across models. Local explanations are generated using LIME to examine individual predictions and assess the stability of feature influence across the sample. Partial dependence analysis is further employed to visualize the marginal effect of key features and to capture nonlinear relationships learned by ensemble and boosting models. The combined use of these approaches allows interpretability results to be validated across complementary explanation techniques.

Model evaluation extends beyond standard accuracy-based metrics to include sensitivity specificity and inference time analysis. All models are trained using identical preprocessing pipelines, including feature scaling, categorical encoding, and class imbalance handling, ensuring that performance differences arise from model characteristics rather than preprocessing inconsistencies. Inference latency is explicitly measured to assess feasibility for real-time or near-real-time clinical deployment.

Through this integrated design, the proposed framework demonstrates that high predictive performance and interpretability can be achieved simultaneously. The results show consistent identification of clinically relevant metabolic

and physiological features as dominant contributors to diabetes risk across models and explanation methods. By aligning probabilistic prediction explainability and deployment-oriented evaluation within a unified framework, this study provides a transparent and reproducible approach for personalized diabetes risk stratification.

2 Literature Review

The growing global impact of type II diabetes has intensified the need for reliable methods to identify individuals at elevated risk before clinical onset[1, 2]. Traditional statistical methods, incorporating logistic regression and Cox proportional hazards models, have been extensively utilized to predict diabetes risk based on demographic, anthropometric, and lifestyle variables such as age, body mass index (BMI), family background, and physical activity[3]. Although these models demonstrate interpretability, they frequently struggle to capture complex, nonlinear relationships and interactions among multiple risk variables[3].

Recent developments in machine learning have improved predictive accuracy and adaptability for modeling diverse clinical data. Tree-based algorithms strike a balance between performance and interpretability through feature importance measures, whereas gradient boosting and neural networks capture complex nonlinear interactions with higher accuracy but lower transparency[4, 5]. Explainable AI methods such as SHAP and LIME allow patient-level interpretation by showing individual feature contributions[6, 7] and rule-based surrogate models extract human-readable decision paths from complex models without much loss in predictive power[8]. Recent work has emphasized risk stratification rather than binary classification to categorize individuals into medium- and high-risk groups[9, 10]. Metabolic and physiological features such as blood glucose, BMI, waist circumference, cholesterol levels, blood pressure, and pregnancy-related weight changes consistently emerge as key predictors[2]. However, few studies integrate probabilistic risk stratification, multi-model evaluation, and multi-level explainability within a single unified framework[10]. This study addresses these gaps by combining probability-based risk scoring across multiple machine learning models with both global and local explainability, allowing transparent and reproducible assessment of diabetes risk at individual and population levels.

3 Methodology

3.1 Study Design and Overview

This study develops an interpretable machine learning approach to classify diabetes risk, differentiating individuals at moderate versus elevated risk of complications. Instead of considering it as a simple binary prediction, the study addresses it as a risk stratification problem, enabling early intervention

and clinically meaningful decision-making. The approach combines multiple machine learning algorithms with complementary explainability methods to provide both population-level and individual-level interpretation. The methodological workflow from data input and preparation to modeling, evaluation, explainable AI, and clinical risk stratification is illustrated in figure 1.

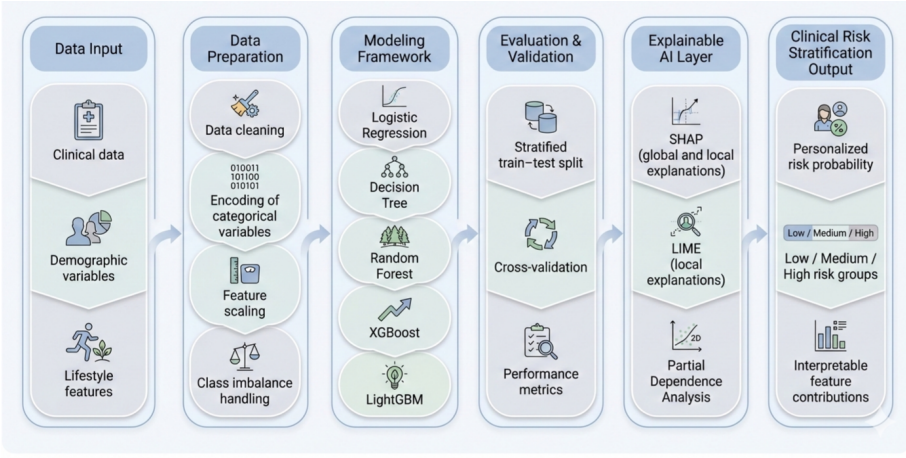


Figure 1: Methodological Workflow Diagram

3.2 Dataset Description

The dataset was sourced from the study[11] and contained approximately 70,000 patient records with 34 demographic, metabolic, and lifestyle features. Unlike commonly used binary labeled datasets, it included multiple clinically distinct subtypes, enabling a risk stratification formulation grounded in real-world clinical heterogeneity. The elevated risk group comprised Type 1 Diabetes, Steroid Induced Diabetes, LADA, Neonatal Diabetes Mellitus and Wolfram Syndrome conditions unified by the need for active clinical management due to autoimmune, pharmacological or congenital metabolic dysfunction [2, 9], while the moderate risk group consisted of prediabetic cases for whom early lifestyle intervention remains the primary clinical tool. This stratification mirrors real world triage logic distinguishing patients requiring immediate intervention from those in a watchful monitoring phase. Records that did not map reliably to either category were excluded and risk levels were binarized for supervised modeling.

3.3 Feature Engineering and Preprocessing

Features were divided into numerical and categorical groups. Numerical variables were standardized and categorical variables were encoded. All preprocessing steps were integrated into Scikit-learn pipelines to ensure consistent

transformations, prevent data leakage, and maintain reproducibility. Class imbalance was addressed using class weighting for most models and a scale pos weight adjustment for XGBoost, improving sensitivity to elevated-risk cases, which is critical in healthcare applications.

3.4 Model Development

Five supervised machine learning algorithms were evaluated; Logistic Regression, Decision Tree, Random Forest, XGBoost and LightGBM. Data were split 80:20 with stratification and cross-validation was performed exclusively on the training partition using 5 fold stratified cross-validation to prevent leakage from the held out test set. Hyperparameters were set to reduce overfitting risk. Tree depth was constrained to a maximum of 3 for Logistic Regression and XGBoost and 4 for Random Forest and LightGBM, minimum samples per leaf were set to 10 across all tree-based models, and learning rates were reduced to 0.05 for XGBoost and LightGBM. LightGBM was further constrained to 20 leaves, and both boosting models applied stochastic subsampling of 0.8 for rows and columns. Class imbalance was addressed using class weighting for all models except XGBoost, which used scale-pos-weight computed as the ratio of negative to positive training samples. Performance metrics included accuracy, precision, recall, F1 score, specificity, and ROC AUC, and inference latency was measured at the batch level to assess deployment feasibility.

3.5 Risk Stratification via Predicted Probabilities

For top performing models, the predicted probability of the elevated risk class was used to generate a continuous, personalized risk score for each patient. Patients were stratified into three categories; Low (probability less than 0.3), Medium (0.3 less than or equal to probability less than 0.7) and High (probability more than or equal to 0.7) as illustrated in Figures 2 and 3. Thresholds of 0.3 and 0.7 were selected to reflect clinical decision boundaries consistent with probability based risk stratification in prior diabetes modeling work [9, 10], balancing sensitivity in identifying high risk cases against the risk of over referral in lower risk populations.

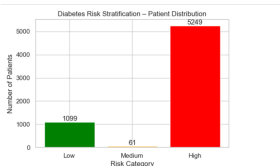


Figure 2: Risk Categorization

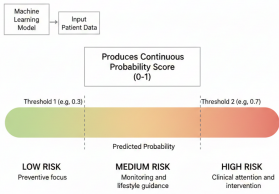


Figure 3: Risk Stratification Framework Diagram

3.6 Model Explainability Framework

To improve transparency and support clinical interpretability, multiple complementary explainability methods were implemented.

SHAP quantified global and local feature contributions using LinearExplainer for Logistic Regression and TreeExplainer for tree-based models. LIME provided local explanations for individual predictions while partial dependence plots visualize the marginal effects of the top features on predicted risk. Model performance and interpretability were supported through ROC curves, confusion matrices, risk distribution plots, SHAP and LIME summaries, and partial dependence plots. This combination of probability-based scoring and multi-method explainability is a major technical contribution, enabling interpretable and clinically relevant insights.

3.7 Reproducibility and Implementation

All experiments were conducted in Python using Scikit learn, XGBoost, LightGBM, SHAP and LIME. Random seeds were fixed for reproducibility, and the modular pipeline supports transparent experimentation and future extension to external validation or clinical studies.

4 Results and Discussion

4.1 Overall Model Performance

All five supervised learning algorithms performed well in stratifying diabetes risk. Ensemble and boosting methods outperformed linear and single-tree models by capturing nonlinear interactions among demographic and metabolic features. LightGBM achieved the highest performance with a test accuracy of 0.9988 and ROC AUC of 0.9996, correctly classifying all 1,075 medium risk cases and 5,321 out of 5,334 high risk cases as shown in figure 4. XGBoost and Random Forest each exceeded 0.97 accuracy and 0.98 ROC AUC while Logistic Regression achieved 0.931 accuracy and Decision Tree showed slightly lower generalization. Full metrics are summarized in Table 1.

The strong performance metrics reflect both the clinical structure of the dataset and the rigor of the evaluation design. The included subtypes present with physiologically differentiated biomarker profiles, producing meaningful natural separability in the feature space that the models learned reliably rather than an artifact of modeling choices. The test set comprising 20 percent of the data was sealed prior to any model training, cross-validation or hyperparameter configuration ensuring reported metrics reflect performance on completely unseen patients. Preprocessing transformations were fit exclusively within the training partition, structurally preventing data leakage and cross-validation was conducted on the training partition only. As shown in Figure 5, cross-validation and test ROC AUC scores are closely aligned across all five models, with standard deviations ranging from 0.0001 to 0.0012, confirming stable gen-

eralization rather than overfitting or evaluation bias.

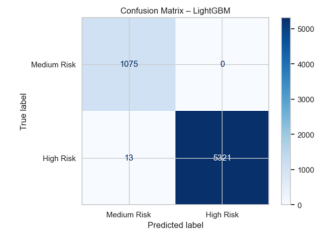


Figure 4: Confusion Matrix for LightGBM

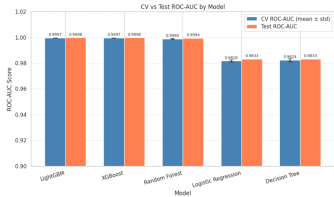


Figure 5: CV vs. Test ROC AUC by Model

Table 1: Predictive Performance of Machine Learning Models for Diabetes Risk Stratification

Model	Acc.	F1	ROC AUC	Prec.	Rec.	Spec.
Logistic Regression	0.931	0.957	0.983	0.991	0.925	0.958
Decision Tree	0.972	0.983	0.983	1.000	0.967	1.000
Random Forest	0.972	0.983	0.999	1.000	0.967	1.000
XGBoost	0.992	0.995	1.000	1.000	0.990	1.000
LightGBM	0.998	0.999	1.000	1.000	0.998	1.000

4.2 Sensitivity, Specificity, and Deployment Feasibility

Precision values were high across all models, indicating low false positive rates. Ensemble and boosting algorithms achieved near-perfect specificity, while Logistic Regression showed slightly lower specificity due to overlapping class distributions. Recall remained above 0.997 for nonlinear models, ensuring strong sensitivity in identifying individuals at elevated risk. Inference analysis showed all models produced predictions within sub-second times at the batch level. Logistic Regression and Decision Tree were the fastest because of their simpler structures. LightGBM and XGBoost were efficient among the ensembles, whereas Random Forest was slower due to its larger ensemble size. Considering both performance and efficiency, LightGBM provided the best balance for deployment.

4.3 Probability-Based Risk Stratification

Predicted probabilities from optimal models enabled personalized risk scores and risk bins. The probability distributions showed clear separation between the moderate and elevated risk groups, with LightGBM producing stable, well-separated estimates. This approach allows both patient-level interpre-

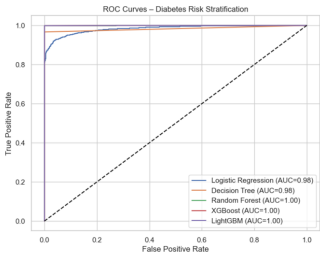


Figure 6: ROC Curve

tation and population-level comparison across models.

4.4 Global Feature Importance: SHAP Analysis

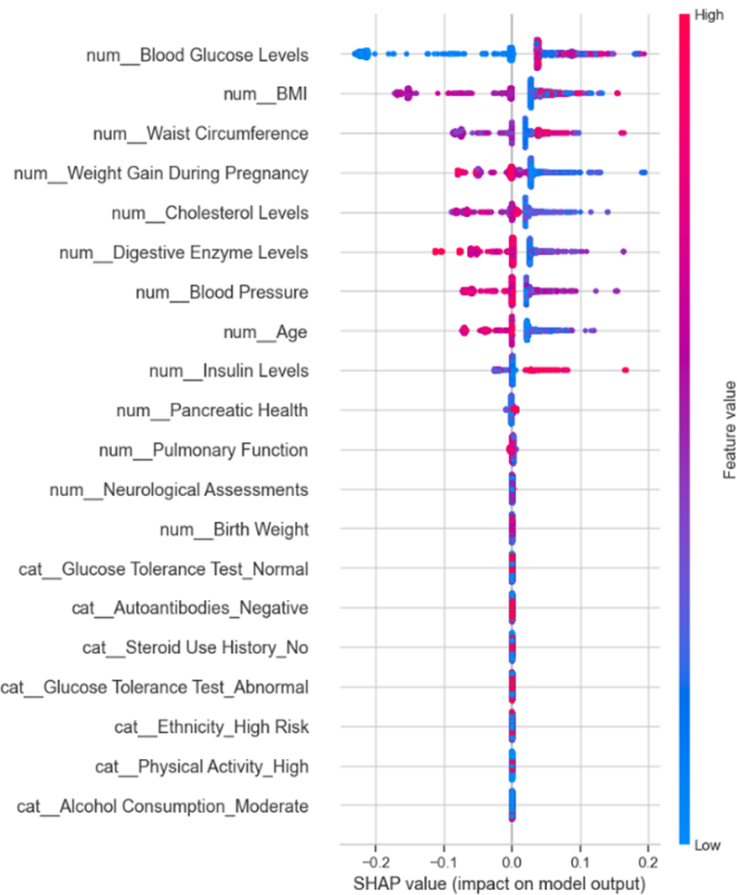


Figure 7: SHAP Values-LightGBM

SHAP analysis revealed consistent patterns of importance across LightGBM, XGBoost, and Random Forest. Blood glucose emerged as the most influential predictor, followed by BMI, cholesterol, waist circumference, age, and pregnancy-related weight indicators. For clarity, SHAP visualizations are presented for the LightGBM model, as similar patterns were observed across other ensemble models.

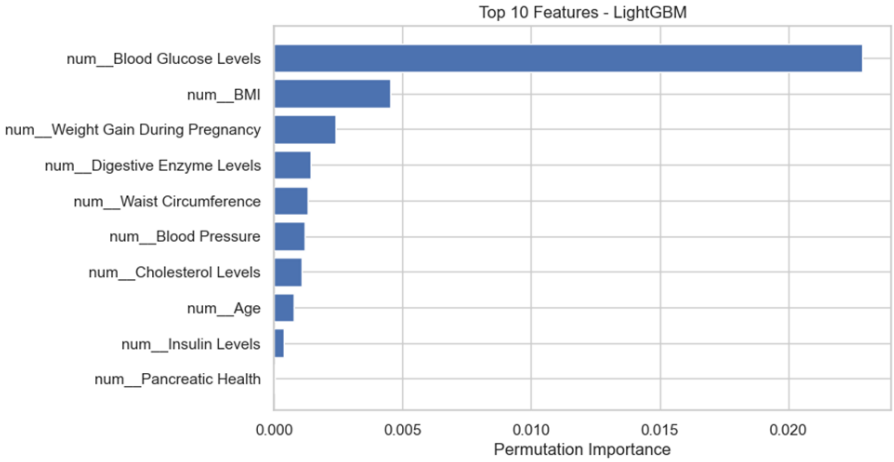


Figure 8: LIME Explainability: LightGBM

4.5 Local Explainability: LIME Analysis

LIME provided individualized explanations showing how features contributed to single predictions. For ensemble models, blood glucose, BMI, waist circumference, cholesterol, and pregnancy indicators were consistently important, whereas Logistic Regression highlighted extreme glucose values and elevated blood pressure, and the Decision Tree reflected threshold-based decisions with BMI and glucose splits having significant contributions.

4.6 Partial Dependence Analysis

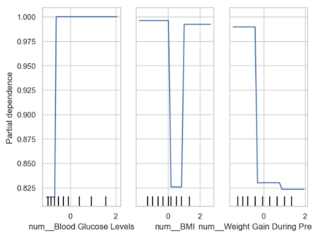


Figure 9: Partial Dependence Plot

Partial dependence plots visualize the marginal effect of blood glucose, BMI, and gestational weight gain on predicted risk probability. Sharp threshold based transitions are observed for all three features with predicted probability dropping significantly at specific standardized values. These non-linear relationships confirm that ensemble models learned clinically meaningful decision boundaries rather than linear approximations and are consistent with the feature importance patterns identified through SHAP and LIME analysis.

4.7 Summary of Key Findings and Discussion

Ensemble and boosting models outperformed linear and single tree approaches across all evaluation metrics with LightGBM providing the best balance between predictive accuracy and computational efficiency. Metabolic features including blood glucose, BMI, cholesterol, waist circumference and gestational weight gain consistently drove elevated risk predictions across all five models and all three explainability methods. When SHAP, LIME and partial dependence analysis independently identify the same predictors, it provides strong evidence that these features represent genuine clinical signals rather than artifacts of any single modeling approach, and their alignment with established clinical knowledge of diabetes risk [2] further supports the validity of the framework.

The technical contribution of this work lies in combining probabilistic risk stratification with interpretable machine learning within a single reproducible pipeline. By moving beyond binary classification to assign individualized risk scores, the framework directs low-risk patients toward preventive monitoring, medium-risk patients toward lifestyle intervention and high-risk patients toward immediate clinical evaluation. The use of model appropriate explainers specifically LinearExplainer for Logistic Regression and TreeExplainer for tree-based and boosting models ensures explanations are technically consistent with each model’s internal decision logic rather than treated as a post-hoc addition.

The primary limitation is the use of a single dataset source without external validation. Future work incorporating independent cohorts, longitudinal follow up and multimodal data would further establish the generalizability of the framework toward clinical deployment.

5 Conclusion

This study establishes that explainable machine learning can deliver clinically interpretable and actionable diabetes risk stratification. The alignment of model-identified predictors with recognized clinical risk factors supports the framework’s potential to guide early intervention, enhance patient outcomes, and enable population-level preventive strategies.

References

- [1] World Health Organization. *Diabetes Fact Sheet*. <https://www.who.int/news-room/fact-sheets/detail/diabetes>. Accessed 2025. 2021.
- [2] Yan Zheng, Sylvia H. Ley, and Frank B. Hu. “Global Aetiology and Epidemiology of Type 2 Diabetes Mellitus and Its Complications”. In: *Nature Reviews Endocrinology* 14 (2018), pp. 88–98.
- [3] Peter E. H. Schwarz, Jie Li, et al. “Diabetes Risk Prediction Models: Traditional Statistical Approaches and Their Limitations”. In: *Diabetes Research and Clinical Practice* 97 (2012), pp. 1–12. DOI: 10.1016/j.diabres.2012.01.002.

- [4] Ioannis Kavakiotis, Olga Tsave, Athanasios Salifoglou, et al. “Machine Learning and Data Mining Methods in Diabetes Care”. In: *Computational and Structural Biotechnology Journal* 15 (2017), pp. 104–116. DOI: 10.1016/j.csbj.2016.12.005.
- [5] Jerome H. Friedman. “Greedy Function Approximation: A Gradient Boosting Machine”. In: *Annals of Statistics* 29 (2008), pp. 1189–1232. DOI: 10.1214/aos/1013203451.
- [6] Scott M. Lundberg and Su-In Lee. “A Unified Approach to Interpreting Model Predictions”. In: *Advances in Neural Information Processing Systems*. 2017.
- [7] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. “Why Should I Trust You? Explaining the Predictions of Any Classifier”. In: *Proceedings of the ACM SIGKDD Conference*. 2016.
- [8] George Popescu et al. “Rule Extraction from Support Vector Machines: Survey and New Methods”. In: *IEEE Transactions on Neural Networks* 19.2 (2008), pp. 215–225. DOI: 10.1109/TNN.2007.914318.
- [9] X. Li, F. Ding, L. Zhang, et al. “Interpretable Machine Learning for Prediabetes Risk Prediction”. In: *BMC Public Health* (2025). DOI: 10.1186/s12889-025-22419-7.
- [10] M. Li, A. Tan, L. Song, et al. “Performance of Diabetes Risk Prediction Models: A Systematic Review and Meta-Analysis”. In: *Endocrine Connections* (2025).
- [11] Radwa Ahmed Osman. “Explainable AI-Driven 1D-CNN with Efficient Wireless Communication System Integration for Multimodal Diabetes Prediction”. In: *AI* 6.10 (2025), p. 243. DOI: 10.3390/ai6100243. URL: <https://www.mdpi.com/2673-2688/6/10/243>.

The Impact of Early College and Dual-Enrollment Courses on Success in Upper Level Undergraduate Computer Science Courses*

James A. Jerkins and Daniel Ray
Computer Science and Information Systems
University of North Alabama
Florence, AL 35632
`{jajerkins, dray4}@una.edu`

Abstract

This paper provides results from a longitudinal study of the observed effects of early college experience among a population of Computer Science students at a mid-sized regional university. An anticipated advantage in average outcome for students with early college credit over those without in CS1 is established and quantified. An attempt is made to trace that advantage and provide information about the durability of that advantage. The results of the paper are useful for informing future advancements in early college programs.

1 Introduction

Early college programs have been offered in various forms since the 1950s. Research into the efficacy of early college programs that provide college credit for high school students prior to high school graduation has been a prominent topic for decades. However, research has been complicated by the fact that early college programs were initially created in an uncoordinated fashion with varying language and terminology to describe them. Descriptive terms of art like “dual enrollment”, “dual credit”, “concurrent enrollment”, and “early college high school” are often conflated among planners, executors of programs, and

*Copyright ©2026 by the Consortium for Computing Sciences in Colleges. Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the CCSC copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Consortium for Computing Sciences in Colleges. To copy otherwise, or to republish, requires a fee and/or specific permission.

researchers in the field. In this paper we will prefer the term “early college” but intend that terminology as an umbrella that includes all forms of these programs.

A key idea common to early college programs since their inception is that talented high school students may be more likely to pursue and complete college degrees if they are given the opportunity to acclimate to the demands of college level coursework prior to college. The programs are also based on the idea that high school students will be more likely to undertake college level courses while still in high school if they can earn college credit. Researchers have concluded that for many students this has been historically true as well as documenting many benefits of early college experiences.

The authors observed in their CS courses that students struggling in upper level courses often had earned high grades in lower level CS courses. One possible explanation for the difference could be that early college experiences do not adequately prepare students for the low-structure, high-rigor environment of upper level CS courses. In this paper we investigate the long-term effect of early college for CS students and attempt to determine if the documented benefits are enduring.

2 Background

The popularity of early college courses has been increasing with over 2.8 million U.S. high school students participating in the 2023-24 school year [1]. A large body of existing research has documented how students with early college experience perform in introductory college course work in various disciplines including computer science. Researchers have argued that comparing the grades of students without early college experience to those with early college experience best highlights the influence of early college programs. Researchers have routinely concluded that early college experiences benefit students in introductory college courses.

Students who took early college courses were more likely to attend college, more likely to persist in college course work, and more likely to take a full load in college [2]. Early college experiences have been shown to broaden participation in computer science among low-income students [3]. Researchers in South Carolina investigated the impact of early college experiences on students and concluded that students with early college were more likely to be retained and more likely to graduate in six years [4]. Grubb et al. studied how early college experiences influenced remediation and graduation rates from a two year college. They concluded that students with early college experiences were 3 times less likely to take remediation and almost 2.5 times more likely to graduate in two years [5].

Research has also established that how and where early college experiences take place as well as characteristics of students taking the early college courses

have an impact on outcomes [6, 7, 8]. Weston et al. found that, among female students, programming experiences during high school, and taking the CS AP exam were among the best predictors of persistence 3 years on in both CS and other tech-related fields [9].

Burgiel et al. conducted a wide-ranging study of the content and pedagogy of high school CS courses in relation to student success in introductory college CS courses [10]. They found that the only aspect of the pedagogy that was correlated with student success in these courses was an emphasis on coding and a solid background in mathematics. Chen et al. evaluated the role of AP courses in computer science and mathematics as a predictor of student success in introductory college CS courses compared to non-AP classes and students who did not take such classes [11]. They found evidence that experience in either AP calculus or AP computer science had similarly strong positive effects on student's grades in introductory computer science classes.

There is perhaps no academic area where an interest in the success of early college has been more studied than in computer science. However, the long-term effects of early college experiences as students progress through undergraduate programs have been left unstudied. In this study, we examine course grade outcomes as students progress through an undergraduate CS program by comparing two groups, students with any reported early college experience, and students without early college. Specifically, we investigate the following three hypotheses:

1. Students who have early college experience perform better than those without early college in the introductory CS course.
2. The performance advantage apparent in the introductory course of students with early college experience over those without persists through the end of their sophomore level CS courses.
3. In upper level CS courses, early college experience is correlated with lower grade outcomes when compared to students with no early college experience.

We leave to other researchers the task of picking apart the different effects of various early college delivery structures, curricular approaches, impact on broadening participation, and student persistence or retention. In the remainder of this paper we present, analyze, and discuss data from two groups of students, those with early college credit and those with no early college credit.

3 Methodology

Our initial study population ($n = 425$) contained any student over the past five years at our mid-size regional university who at any point declared a major in computer science. For all students our university tracks awarded early college credits (regardless of early college program type) as well as any awarded AP credits. For the purposes of this study we lump all early college credit and AP

credit into a total number of early college credits per student as our indicator of early college experience.

In order to eliminate students lacking any meaningful progression toward completion of a CS degree we removed all students from the population that did not have a graded attempt in the introductory programming course (CS1). That left a remaining population of size $n = 270$.

For each student, we collected the last semester that they were enrolled in any undergraduate course at our university. This allowed us to divide the population into five cohorts; 2021, 2022, 2023, 2024, and 2025. For example, the 2021 cohort includes students whose last undergraduate course was recorded in the 2020-2021 academic year. The 2021 cohort includes the Fall 2020, Spring 2021, and Summer 2021 semesters. The other cohorts are organized in a similar fashion for their respective academic years.

The set of students who graduated with an undergraduate degree in CS in a given year appear as a subset of students in that cohort year. Each cohort also contains CS majors who took their most recent undergraduate course in that year but did not graduate. Finally, each cohort also contains students who at some point changed majors from CS to some other major and took their most recent undergraduate course during the cohort year, whether or not they graduated.

Each cohort year was divided into two categories; those students with recorded early college credit, and those without. Table 1 gives the initial size of each category (the number from each group with a recorded grade in the introductory programming course) in each cohort year.

Table 1: CS1 students per cohort year

	2021	2022	2023	2024	2025
Early College Credit	7	12	6	12	16
No Early College Credit	44	32	37	41	63
Total	51	44	43	53	79

For every student in our data set we collected letter grades in a selection of required CS courses at the introductory, sophomore and upper level. Specifically, we collected grades for the introductory programming course (CS1), two sophomore level courses; CS2, and Data Structures and Algorithms (CS3), and four upper level required courses. We selected the upper level courses as representative of rigorous advanced CS courses from a diverse set of topic areas; specifically, programming languages, operating systems, networking, and a first semester in software engineering. It is useful to note that CS1, CS2, and CS3 in our program are taken as a sequence (routinely in consecutive semesters), while the upper level courses may be taken in any order after successfully completing CS3.

We coded the letter grade for each student in each course by assigning a quality point value of 4 for an A, 3 for a B, 2 for a C, 1 for a D and 0 for any other outcome (F or W). For each class the average of the quality points was computed in each cohort year, for both the early college and no early college categories, producing a total of 70 discrete data points. Naturally, as each cohort progressed through the curriculum, the number of students taking courses in the major decreased through academic attrition. The quality point value averages only included students where an attempt was made and ignored students without attempts.

Using this data our study supports, first, the establishment of a measure of the advantage in course outcomes for students with early college credit over students without early college in CS1. This aligns with existing research reports. Second, this data supports the tracing of that same advantage through sophomore class outcomes. Lastly, it supports the tracking of any deterioration in the advantage enjoyed by students with early college credit as students progress through upper level degree courses. This data was then examined to determine support for our hypotheses.

4 Findings

As anticipated by the existing literature, our study found a significant positive advantage in outcomes for the introductory CS course (CS1) for students with early college credit over those without. For each of the five cohort years the average quality points attained by students with early college credits was significantly higher than that attained by students without early college credits. For example, in the 2025 cohort the average quality points earned in the introductory programming course by students with early college credits was approximately 2.875 versus 2.365 for those without. The delta between these two values is 0.51, meaning that on average the 2025 cohort students with early college credit performed more than half a letter grade better than those without early college.

Table 2 shows the delta value of average quality points in CS1 between students with and students without early college credit for all five cohort years. This data suggests we can accept Hypothesis 1; students who have early college experience perform better than those without early college in the introductory programming course.

Table 2: Delta in average quality points between students with/without early college

	2021	2022	2023	2024	2025
Average Outcome Delta	0.64	0.22	1.28	0.6	0.51

Table 2 establishes a baseline for the magnitude of the positive advantage in average course outcome enjoyed by students with early college credit over those without early college credit. Using this baseline, we traced this advantage

through the completion of sophomore level courses.

To facilitate tracing, we calculated the average number of quality points for both categories of students in each of the five cohort years for two different sophomore level CS courses; Computer Science II (CS2) and Data Structures and Algorithms (CS3). The delta between the average quality points outcome for students with early college credit and those without for these two classes is shown in Table 3. The table also compares the delta in quality points between the two categories from each of CS2 and CS3 with the baseline delta established in CS1.

Table 3: Persistence of the CS1 advantage in sophomore level courses

	2021	2022	2023	2024	2025
CS2 Delta	0.836	-0.09	0.2	0.39	0.25
Change in Delta from CS1 to CS2	0.19	-0.3	-1.08	-0.21	-0.25
CS3 Delta	0.44	-0.38	0.85	0.44	-0.93
Change in Delta from CS1 to CS3	-0.2	-0.6	-0.43	-0.16	-1.44

Contrary to the common expectation, the benefit from early college experiences present in CS1 rapidly deteriorated and sometimes entirely inverted when analyzing sophomore level course outcomes. Course outcomes for CS2 are particularly surprising because these two classes (CS1 and CS2) are routinely taken in consecutive semesters. For four out of five cohort years the advantage present in CS1 diminished drastically in CS2. As shown in the second row of Table 3 the size of the positive advantage in course outcomes in those four years generally diminished between a quarter to a third of a letter grade as compared to the advantage observed in CS1. For the 2023 cohort year, the effect diminished by over a full letter grade. That is, students in that cohort year with early college credit outperformed students without early college credit by over a full letter grade **less** than they had in CS1. In the 2022 cohort year the no early college students actually outperformed those with early college experience overall when they had been outperformed by nearly a quarter of a letter grade in CS1.

The analysis surrounding CS3 outcomes exhibited a similar trend. It is important to note students who do not pass CS2 are not allowed to attempt CS3 and are not included in the CS3 averages. The absence of these lower scoring students in CS3 tend to make the overall outcome higher. Despite this, we observed for all five of the cohort years there was a decrease in the net positive advantage in outcomes for students with early college credit when comparing CS1 outcomes with CS3 outcomes. Furthermore, for three of the five cohort years the decrease in net effect continued to deteriorate from CS2 levels despite the likely absence of the poorest scoring students.

The data analysis supports a rejecting of Hypothesis 2. The performance

gap apparent in the introductory course between students with early college experience and those without **does not** persist through the end of the sophomore level CS courses and instead precipitously diminishes.

We found similar, but less compelling evidence of further degradation in the advantage enjoyed by students with early college credit when comparing outcomes in upper level courses with the baseline effect observable in CS1. For each of the four upper level courses and each cohort year, the delta of average quality points was computed for students with and without early college credit. Table 4 shows the difference between those values and the delta established in the CS1 baseline. Negative values indicate that the advantage in quality point outcomes between the group with early college credits and the group without early college credit was smaller than the advantage observed between those same two groups in CS1. Negative values suggest a diminishment of the effect of early college experience on course outcomes overall.

Table 4: Difference in the delta of averages between categories compared to delta of averages in CS1

	2021	2022	2023	2024	2025
Programming Languages	-0.5	-0.05	-0.02	-0.47	-0.64
Operating Systems	0.19	-0.3	-1.08	-0.21	-0.25
Networking	0.44	-0.38	0.85	0.44	-0.93
Software Engineering	-0.2	-0.6	-0.43	-0.16	-1.44

From the twenty data points (4 classes each over 5 cohort years) we identified four occasions (20%) where the no early college credit group outperformed the early college group. Moreover, as Table 4 shows it was common for the early college group to have lower performance compared to other students in upper level courses than the baseline delta shown in CS1. In 80% of instances (16 of 20) the net advantage in outcome showed at least some degradation.

Overall, the magnitude of the degradation of the advantage shown in upper level courses as compared to the CS1 baselines is not as significant as that observed in sophomore classes. For this reason Hypothesis 3 is rejected. We did not find strong evidence that early college experience is correlated with lower grade outcomes when compared to students with no early college experience in upper level CS courses. Neither did we find that early college experience was well associated with better overall performance as compared to students without it in these same classes.

Our analysis showed that for all of the upper level courses participation from each cohort year decreased by at least 50% from its initial size and for some courses by significantly more (sometimes 85 to 90%). In these upper level courses half or more of the original cohort had been lost to academic attrition. Analysis showed that both categories (early college and no early college) ex-

perienced similar attrition rates. Overall, less academically successful students fell away and more successful students remained. This constrained our ability to know how the lower performing students in each category would have performed in these classes had they taken them. The dependence of upper level course outcomes on sophomore course outcomes complicated our findings leaving an opportunity for further investigation of Hypothesis 3.

5 Conclusion

Our study found that a clear advantage was enjoyed in the outcome of an introductory programming course by students with early college credit over those without. However, we also found that the advantage severely degraded by the time these same students completed their sophomore courses and that it did not meaningfully reappear and often continued to shrink as they progressed through upper level CS course work.

What could explain the consistent, sharp decline in the magnitude of the advantage in course outcomes enjoyed by students with early college credit over those without? Is it that the advantage in course outcomes to early college students is being erased as other students accrue hard earned lessons and academic skills as they progress through the curriculum? This study is not prepared to answer these questions and only further research seems likely to answer them with any certainty.

However, one possibility that could explain the apparently paper thin depth of the advantage from early college experience is that students in early college are being prepared with foundational content knowledge which serves them in their introductory courses, but may not be receiving adequate preparation for the rigorous, personal academic effort expected of students in college. Additionally, the content knowledge they do acquire in early college courses could create a false confidence that they are prepared for college. The effects of such a disadvantage would likely first be apparent in sophomore course work. While such an explanation amounts to no more than speculation on our part, it could explain the precipitous decline in advantage in sophomore classes as well as the continued declines shown in upper level courses.

Regardless, the data provides a foundation and motivation for planners and executors of early college programs to review their procedures and to ensure that their students are not just knowledgeable but also outfitted with the skills and personal attributes necessary for low structure, high rigor college courses.

References

- [1] Columbia University. *High School Dual Enrollment Grows to 2.8 Million* — *ccrc.tc.columbia.edu*. <https://ccrc.tc.columbia.edu/easyblog/high-school-1-dual-enrollment-grows.html>. [Accessed 09-01-2026]. Sept. 2025.
- [2] Kristen P. Kremer. “Predictors of College Success Outcomes in Emerging Adults: The Role of High School Dual Enrollment Courses”. In: *Emerging Adulthood*

- 10.1 (2022), pp. 188–196. DOI: 10.1177/2167696820916639. eprint: <https://doi.org/10.1177/2167696820916639>. URL: <https://doi.org/10.1177/2167696820916639>.
- [3] Brian P. An. “The Impact of Dual Enrollment on College Degree Attainment: Do Low-SES Students Benefit?” In: *Educational Evaluation and Policy Analysis* 35.1 (2013), pp. 57–75. DOI: 10.3102/0162373712461933. eprint: <https://doi.org/10.3102/0162373712461933>. URL: <https://doi.org/10.3102/0162373712461933>.
- [4] Michael Tucker, Jonathan Poon, and Sophie Greenwell. “Dual Enrollment, First-Year Retention, and Graduation: Analyzing the Impact of Dual Enrollment on Student Success Outcomes at a Public Research University”. In: *Journal of Higher Education Theory and Practice* 23.9 (Mar. 2023). DOI: 10.33423/jhetp.v23i9.6132. URL: <https://articlegateway.com/index.php/JHETP/article/view/6132>.
- [5] John M. Grubb, Pamela H. Scott, and Donald W. Good. “The Answer Is Yes: Dual Enrollment Benefits Students at the Community College”. In: *Community College Review* 45.2 (2017), pp. 79–98. DOI: 10.1177/0091552116682590. eprint: <https://doi.org/10.1177/0091552116682590>. URL: <https://doi.org/10.1177/0091552116682590>.
- [6] Mark M. D’Amico et al. “Dual Enrollment Variables and College Student Persistence”. In: *Community College Journal of Research and Practice* 37.10 (2013), pp. 769–779. DOI: 10.1080/10668921003723334. URL: <https://doi.org/10.1080/10668921003723334>.
- [7] Grant Clayton, Anna Faria, and Patty Witkowsky. “Concurrent Enrollment Outcomes by Course Location”. In: *Community College Journal of Research and Practice* 49.3 (2025), pp. 153–168. DOI: 10.1080/10668926.2023.2283597. eprint: <https://doi.org/10.1080/10668926.2023.2283597>. URL: <https://doi.org/10.1080/10668926.2023.2283597>.
- [8] Xiaodan Hu and Hsun-Yu Chan. “Does Delivery Location Matter? A National Study of the Impact of Dual Enrollment on College Readiness and Early Academic Momentum”. In: *Teachers College Record* 123.4 (2021), pp. 1–32. DOI: 10.1177/016146812112300401. eprint: <https://doi.org/10.1177/016146812112300401>. URL: <https://doi.org/10.1177/016146812112300401>.
- [9] Timothy J. Weston, Wendy M. Dubow, and Alexis Kaminsky. “Predicting Women’s Persistence in Computer Science- and Technology-Related Majors from High School to College”. In: *ACM Trans. Comput. Educ.* 20.1 (Sept. 2019). DOI: 10.1145/3343195. URL: <https://doi.org/10.1145/3343195>.
- [10] Heidi Burgiel, Philip M. Sadler, and Gerhard Sonnert. “The Association of High School Computer Science Content and Pedagogy with Students’ Success in College Computer Science”. In: *ACM Trans. Comput. Educ.* 20.2 (Apr. 2020). DOI: 10.1145/3381995. URL: <https://doi.org/10.1145/3381995>.
- [11] Chen Chen et al. “High School Calculus and Computer Science Course Taking as Predictors of Success in Introductory College Computer Science”. In: *ACM Trans. Comput. Educ.* 21.1 (Dec. 2021). DOI: 10.1145/3433169. URL: <https://doi.org/10.1145/3433169>.

Soccer Action Detection Using TimeSformer: A Transformer-Based Approach to Video Understanding*

Musfikur Rahaman¹, Robin Ghosh²,
Shucheta Bari³, Abdullah Al Faysal⁴

¹Department of Information Science
University of Arkansas at Little Rock
Little Rock, AR 72204

²Division of Computing Sciences
Arkansas Tech University
Russellville, AR 72801

³School of Business
University of Arkansas at Little Rock
Little Rock, AR 72204

⁴Software Engineer
eTek IT Services Inc.
Schaumburg, IL 60173

mrahaman@ualr.edu, rghosh@atu.edu

sbari@ualr.edu, faysal.iit0313@gmail.com

Abstract

Deep learning-based sports video analysis has advanced rapidly, enabling automated understanding of complex athletic activities across large volumes of multimedia content. Soccer is one of the most widely followed sports globally, yet fine-grained action recognition remains chal-

*Copyright ©2026 by the Consortium for Computing Sciences in Colleges. Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the CCSC copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Consortium for Computing Sciences in Colleges. To copy otherwise, or to republish, requires a fee and/or specific permission.

linging due to subtle motion variations and long-range temporal dependencies. In this work, we present a transformer-based approach for soccer action recognition using the SoccerAct10 dataset, which contains approximately 3,600 annotated video clips spanning ten action classes: corner, foul, free kick, goal kick, long pass, on target, penalty, short pass, substitution, and throw-in. The proposed method employs the TimeSformer (Time-Space Transformer) model, which represents videos as sequences of image patches and applies global self-attention across spatial and temporal dimensions, enabling effective modeling of long-range dependencies without relying on recurrent or convolutional operations. By fine-tuning a TimeSformer model pretrained on large-scale video data, the proposed approach achieves a Top-1 classification accuracy of 91.13% on the validation set. Results are reported from a single experimental run; future work should validate findings across multiple runs to confirm stability. These results demonstrate that transformer-based architectures are highly effective for single-action soccer video recognition and show strong potential for applications in sports analytics, automated broadcasting, and intelligent sports media systems.

1 Introduction

The rapid growth of digital sports media, including live broadcasts, recorded matches, highlights, and social platforms, has increased the need for automated systems that can accurately analyze sports video. In soccer, *fine-grained action recognition* focuses on distinguishing visually similar but semantically distinct events such as pass types, set pieces, fouls, and substitutions. This task is challenging due to complex spatiotemporal dynamics, camera motion, and high visual similarity between action categories [1, 2]. Traditional handcrafted-feature methods and CNN-based approaches have shown promise but often struggle to capture long-range temporal dependencies present in broadcast videos [3, 4].

To address these limitations, this study employs TimeSformer, a transformer-based video recognition model that uses self-attention to jointly model spatial and temporal information across video clips [5]. We evaluate whether transformer-based architectures improve fine-grained soccer action recognition on the SoccerAct10 dataset compared with prior CNN- and recurrent-based baselines [1, 5, 6]. Using SoccerAct10 (10 actions: corner, foul, free kick, goal kick, long pass, on target, penalty, short pass, substitution, and throw-in), we fine-tune the model and report Top-1 accuracy, confusion matrices, precision, recall, and F1-score [1, 7, 2].

Automated soccer action recognition has practical value for coaching and tactical analysis, real-time highlight generation, and structured performance analytics [8, 2]. Prior work shows that reliable action recognition supports downstream tasks such as event detection and summarization [9]. Recent advances in transformer-based video models further motivate this work, as at-

tention mechanisms enable modeling of longer temporal context [10, 5]. Given TimeSformer’s strong benchmark performance using divided space–time attention [5], this study investigates its effectiveness and limitations for fine-grained soccer action recognition on SoccerAct10 [7, 6].

2 Literature Review

Early sports video analysis relied on handcrafted visual and motion descriptors such as Histogram of Oriented Gradients (HOG), Scale-Invariant Feature Transform (SIFT), optical flow, and edge-based features, combined with traditional machine learning models including Support Vector Machines (SVMs), Hidden Markov Models (HMMs), and autoassociative neural networks [11, 12]. While effective in controlled settings, these approaches struggled with real-world broadcast sports videos due to camera motion, illumination changes, replay overlays, and limited spatiotemporal representation, particularly in soccer where actions often appear visually similar.

The adoption of deep learning, especially Convolutional Neural Networks (CNNs), significantly advanced video understanding by enabling automatic feature extraction [13, 14, 15]. However, frame-based CNN pipelines ignored temporal dependencies, motivating 3D Convolutional Neural Networks (3D CNNs) such as C3D and I3D, which jointly model spatial and short-term temporal information [3, 16]. Large-scale datasets like Sports-1M highlighted the importance of pretraining for transfer learning in sports video analysis [17]. Despite their success, 3D CNNs are computationally expensive and limited by fixed temporal receptive fields.

To better capture temporal structure, recurrent neural networks (RNNs), including LSTM and GRU, were integrated with CNN feature extractors in hybrid architectures such as LRCN and ConvLSTM [4, 18]. In soccer-specific studies, Sen and Deb applied CNN–GRU models with transfer learning on SoccerAct10, achieving strong performance for fine-grained actions [1, 19]. Benchmark datasets such as SoccerNet and SoccerNet-v2 further enabled research on action spotting and event detection in realistic broadcast settings [2, 20]. Additional work explored shot and scene classification to support soccer video summarization [21, 22].

Recent research has shifted toward end-to-end systems for soccer video understanding and highlight generation [8]. Other approaches combined 3D CNNs with bidirectional recurrent networks for special action recognition in football [19]. Surveys on deep learning for video classification emphasize the move toward deep spatiotemporal modeling while noting limitations in capturing long-range dependencies [7].

Transformer-based architectures have emerged as a powerful alternative due to their ability to model global temporal dependencies through self-attention. Vision Transformers (ViT) demonstrated competitive performance without

convolutional operations [10], and this paradigm was extended to video through TimeSformer, which achieved state-of-the-art results on benchmarks such as Kinetics-400 and Something-Something v2 [5, 6]. Despite these advances, transformer-based models remain underexplored for fine-grained soccer action recognition, motivating this study’s investigation of TimeSformer on the SoccerAct10 dataset.

3 Methodology

This study presents a transformer-based deep learning framework for fine-grained soccer action recognition using a pre-trained TimeSformer model [5]. TimeSformer employs self-attention mechanisms to jointly model spatial and temporal information in video sequences, enabling effective representation of complex motion patterns commonly observed in broadcast soccer videos. Unlike convolutional or recurrent architectures that rely on local receptive fields or sequential processing, TimeSformer captures global relationships across frames through attention, making it well suited for modeling long-range temporal dependencies [5, 6]. The overall pipeline consists of dataset preparation, video preprocessing, model configuration, supervised training, and quantitative evaluation, as illustrated in Figure 1.

3.1 Dataset and Preprocessing

Experiments are conducted using the SoccerAct10 dataset, which contains approximately 3,600 annotated video clips spanning ten fine-grained soccer action categories [1, 2]. Each clip contains a single dominant action and is manually annotated to ensure accurate labeling. Videos are collected from professional soccer broadcasts and include diverse camera viewpoints, lighting conditions, and game contexts. The dataset is divided into training, validation, and test sets using a 70%:15%:15% split. For compatibility with the TimeSformer input format, eight frames per clip are uniformly sampled and resized to 224×224 pixels. Frames are converted to RGB format and normalized using ImageNet mean and standard deviation values consistent with the model’s pre-training configuration [13]. Clips with fewer than eight frames are padded, resulting in input tensors of shape $(T, 3, 224, 224)$. Pixel normalization is defined as:

$$\text{Normalized Pixel} = \frac{\text{Pixel Value} - \mu}{\sigma} \quad (1)$$

3.2 Model Architecture and Training

TimeSformer is a transformer-based video classification architecture that replaces convolutional and recurrent operations with divided space-time self-attention [5]. Each input frame is partitioned into non-overlapping 16×16 patches and projected into a 768-dimensional embedding space. Learnable spatial and temporal positional embeddings are added to preserve patch location and frame order information. The resulting spatiotemporal tokens are pro-

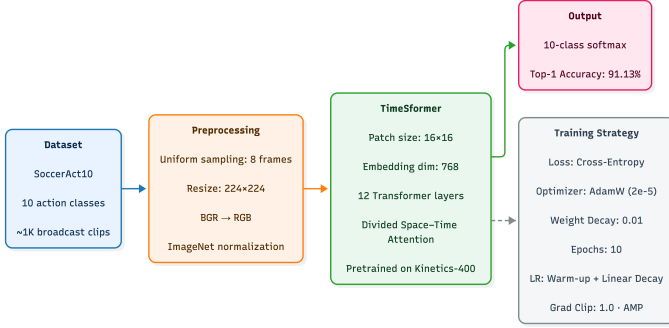


Figure 1: Methodology and Training Strategy of the Proposed TimeSformer-Based Soccer Action Recognition System

cessed by stacked transformer encoder layers, where spatial attention models intra-frame relationships and temporal attention captures dependencies across frames [5, 6]. Self-attention within each transformer layer is computed as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \quad (2)$$

This formulation enables global interaction among spatiotemporal tokens, allowing the model to learn long-range motion patterns and contextual relationships critical for fine-grained action recognition. The architecture is initialized using weights pre-trained on the Kinetics-400 dataset [16, 10]. All experiments were implemented using the PyTorch framework and the HuggingFace Transformers library. The **TimeSformer-Base-K400** model pretrained on Kinetics-400 was fine-tuned on the SoccerAct10 dataset. Training was performed in a supervised manner using the cross-entropy loss function:

$$L = - \sum_{c=1}^C y_c \log(\hat{y}_c) \quad (3)$$

where $C = 10$ denotes the number of action classes. Optimization was carried out using the AdamW optimizer with a learning rate of 2×10^{-5} and weight decay of 0.01. Training progress was monitored using validation accuracy and F1-score to detect potential overfitting. The model was trained for 10 epochs using a batch size of 4 with gradient clipping and a linear warm-up followed by a linear decay learning rate schedule. Automatic mixed precision (AMP) was used when GPU support was available to improve training efficiency. Model performance was evaluated using accuracy, precision, recall, and F1-score, and the checkpoint achieving the highest validation F1-score was selected for final evaluation.

Table 1: Overall Performance Metrics of the TimeSformer Model

Metric	Score (%)
Accuracy	91.13
Precision	91.22
Recall	90.85
F1-Score	90.81

Table 2: Class-wise Precision, Recall, and F1-score for Soccer Action Recognition

Event Class	Precision (%)	Recall (%)	F1-Score (%)
Corner	86.89	91.38	89.08
Foul	86.11	73.81	79.49
Freekick	94.64	85.48	89.83
Goalkick	96.55	98.25	97.39
Longpass	76.27	93.75	84.11
Ontarget	95.31	92.42	93.85
Penalty	94.00	100.00	96.91
Shortpass	97.73	86.00	91.49
Substitution	96.15	94.34	95.24
Throw-in	88.52	93.10	90.76

4 Results and Analysis

4.1 Performance Evaluation

The proposed TimeSformer-based model achieves strong performance on the validation split of the SoccerAct10 dataset, obtaining an accuracy of 91.13%, precision of 91.22%, recall of 90.85%, and an F1-score of 90.81%. These results demonstrate effective recognition of fine-grained soccer actions in broadcast video. The overall performance metrics are summarized in Table 1. It is important to note that these results are based on a single training run. Since deep learning training can involve stochastic variation, future work will report results across multiple runs with mean and standard deviation to provide stronger statistical confidence.

4.2 Class-wise Performance Analysis

Class-wise evaluation shows high accuracy for structured and visually distinctive actions. Goal kicks achieve an F1-score of 97.39%, while penalty kicks reach 100% recall and 94.00% precision (F1-score: 96.91%). Substitution events also perform well with an F1-score of 95.24%, likely due to consistent broadcast cues. Performance decreases for visually similar or context-dependent actions. Long passes achieve an F1-score of 84.11%, while fouls record the lowest F1-score of 79.49%. Detailed class-wise metrics are presented in Table 2.

Table 3: Comparison with Existing Video Action Recognition Approaches

Study / Model	Details
ConvLSTM & LRCN (Rahaman) [23]	Architecture: ConvLSTM and LRCN; Dataset: SoccerAct10; Performance: 70–71% accuracy
Transfer Learning + GRU (Sen & Deb) [1]	Architecture: CNN + GRU; Dataset: SoccerAct10 (earlier version); Performance: ~85–88% accuracy
Football Special Action Recognition (Liu et al.) [19]	Architecture: Deep learning-based; Dataset: Football broadcast videos; Performance: High accuracy (study-specific)
SoccerNet / SoccerNet-v2 Benchmarks [2, 20]	Architecture: Benchmark + baselines; Dataset: SoccerNet / SoccerNet-v2; Performance: mAP-based event detection
C3D (3D CNN) [3]	Architecture: 3D CNN; Dataset: Large-scale action datasets; Performance: Strong short-term performance
I3D (Inflated 3D CNN) [16]	Architecture: 3D CNN; Dataset: Kinetics; Performance: High Top-1 accuracy
TimeSformer [5]	Architecture: Transformer; Dataset: Large-scale benchmarks; Performance: State-of-the-art
Proposed TimeSformer (this study)	Architecture: Transformer; Dataset: SoccerAct10 (~3,600 clips); Performance: 91.13% Top-1 accuracy

4.3 Comparative Analysis with Existing Methods

Table 3 compares the proposed TimeSformer model with representative CNN-, RNN-, and transformer-based approaches used in soccer and general video action recognition tasks. The comparison provides a high-level overview of modeling strategies and typical performance ranges rather than a strictly controlled benchmark. The methods in Table 3 were not re-implemented under identical conditions, and differences in datasets, preprocessing, training configurations, and evaluation protocols may affect performance. For example, Sen and Deb [1] evaluated an earlier, smaller version of SoccerAct10, whereas this study uses an expanded dataset of approximately 3,600 clips. Consequently, the improvement to 91.13% accuracy cannot be attributed solely to the transformer architecture. A controlled re-implementation of baseline models under identical conditions remains an important direction for future work.

Overall, the comparison highlights the advantage of transformer-based models in capturing long-range temporal dependencies through global self-attention. However, fine-grained and context-dependent soccer actions remain challenging across different architectures.

4.4 Training Behavior and Error Analysis

The model shows stable convergence during training over ten epochs. Training loss decreases from approximately 1.68 to near zero, while training accuracy reaches 100% in later epochs. Validation accuracy gradually improves and converges at 91.13%, with the validation F1-score increasing from 66.56% to

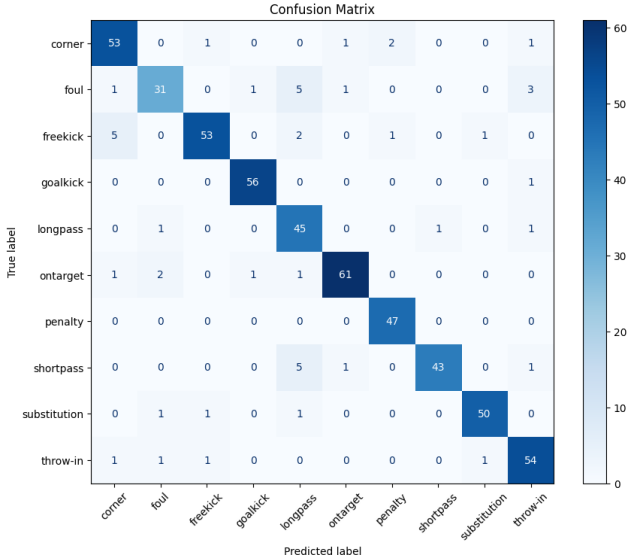


Figure 2: Confusion Matrix of Soccer Action Classification Results

90.81%, suggesting stable generalization without severe overfitting.

Class-level behavior is further illustrated in the confusion matrix shown in Figure 2. Free kicks and corner kicks are occasionally confused due to similar field locations and initial ball placement. Fouls and long passes also show confusion because of overlapping motion patterns and limited contextual cues in RGB video. Short passes are sometimes misclassified as long passes due to camera angle variation and ambiguous distance cues. These results highlight the challenges of fine-grained soccer action recognition.

5 Discussion and Conclusion

This study evaluated a transformer-based architecture for fine-grained soccer action recognition using the SoccerAct10 dataset. The proposed TimeSformer-based model achieved a Top-1 classification accuracy of 91.13%, demonstrating strong performance compared to conventional convolutional and recurrent approaches previously applied to soccer action recognition. These results highlight the effectiveness of representing video data as spatiotemporal patches and leveraging self-attention to capture long-range temporal dependencies that are difficult for CNN- and RNN-based models to model effectively [5, 6].

The observed performance trends align with prior research emphasizing that reliable action recognition is essential for downstream applications such as event detection, highlight generation, and video summarization in broadcast soccer videos [9]. The model performs particularly well on structured and visually distinctive actions, including goal kicks, penalties, and substitutions, which

exhibit stable spatial layouts and predictable temporal patterns.

In contrast, visually overlapping and context-dependent actions such as fouls and the distinction between short and long passes remain challenging. These limitations stem from subtle motion differences, camera viewpoint variation, and the lack of contextual cues beyond RGB video input. As indicated by the comparative analysis, transformer-based architectures provide superior temporal modeling capabilities, but architectural advances alone may not fully resolve ambiguities inherent in fine-grained soccer actions.

Future work may address these challenges by integrating additional modalities such as optical flow, player pose estimation, audio signals, or explicit field-position information to enrich contextual understanding. Exploring longer temporal windows and domain-specific data augmentation strategies may further improve robustness and generalization. Despite current limitations, this study demonstrates the strong potential of transformer-based architectures for soccer video understanding and supports their applicability in sports analytics, automated broadcasting, and intelligent sports media systems.

References

- [1] A. Sen and K. Deb. “Categorization of actions in soccer videos using a combination of transfer learning and gated recurrent unit”. In: *ICT Express* 8.1 (2022), pp. 65–71.
- [2] S. Giancola et al. “SoccerNet: A scalable dataset for action spotting in soccer videos”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2018, pp. 1711–1721.
- [3] D. Tran et al. “Learning Spatiotemporal Features with 3D Convolutional Networks”. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 4489–4497.
- [4] J. Donahue et al. “Long-term Recurrent Convolutional Networks for Visual Recognition and Description”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 2625–2634.
- [5] G. Bertasius, H. Wang, and L. Torresani. “Is space–time attention all you need for video understanding?” In: *Proceedings of the International Conference on Machine Learning (ICML)*. 2021, pp. 813–824.
- [6] A. Arnab et al. “ViViT: A Video Vision Transformer”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 6836–6846.
- [7] A. Rehman and S. B. Belhaouari. “Deep Learning for Video Classification: A Review”. In: *Applied Sciences* 13.3 (2023), p. 2007.
- [8] J. O. Valand et al. “AI-Based Video Clipping of Soccer Events”. In: *Machine Learning and Knowledge Extraction* 3.4 (2021), pp. 990–1008.
- [9] J. Park et al. “Automatic Highlight Generation of Soccer Videos”. In: *Proceedings of the 14th International Conference on Information and Communication Technology Convergence (ICTC)*. 2023, pp. 1867–1871.

- [10] A. Dosovitskiy et al. “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”. In: *International Conference on Learning Representations (ICLR)*. 2021.
- [11] N. Dalal and B. Triggs. “Histograms of Oriented Gradients for Human Detection”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2005, pp. 886–893.
- [12] D. G. Lowe. “Distinctive Image Features from Scale-Invariant Keypoints”. In: *International Journal of Computer Vision* 60.2 (2004), pp. 91–110.
- [13] A. Krizhevsky, I. Sutskever, and G. E. Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 25. 2012, pp. 1097–1105.
- [14] K. Simonyan and A. Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition”. In: *International Conference on Learning Representations (ICLR)*. 2015.
- [15] K. He et al. “Deep Residual Learning for Image Recognition”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778.
- [16] J. Carreira and A. Zisserman. “Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 6299–6308.
- [17] A. Karpathy et al. “Large-scale Video Classification with Convolutional Neural Networks”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014, pp. 1725–1732.
- [18] X. Shi et al. “Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting”. In: *Advances in Neural Information Processing Systems*. Vol. 28. 2015, pp. 802–810.
- [19] S. Wang. “A Deep Learning Algorithm for Special Action Recognition of Football”. In: *Mobile Information Systems* 2022 (2022), p. 6315648.
- [20] A. Delière et al. “SoccerNet-v2: A Dataset and Benchmarks for Holistic Understanding of Broadcast Soccer Videos”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2021.
- [21] M. Rafiq et al. “Scene classification for sports video summarization using transfer learning”. In: *Sensors* 20.6 (2020), p. 1702.
- [22] R. A. Minhas et al. “Shot Classification of Field Sports Videos Using AlexNet Convolutional Neural Network”. In: *Applied Sciences* 9.3 (2019), p. 483.
- [23] Musfikur Rahaman. *League of Learning: Deep Learning for Soccer Action Video Classification*. ATU Scholars Symposium. 2024. URL: https://orc.library.a.tu.edu/atu_rs/2024/2024/17.

Designing a Hardware-Centric Cross-Disciplinary Course in IoT: A Case Study*

David L. Tarnoff
Department of Computing
East Tennessee State University
Johnson City, TN 37614
`tarnoff@etsu.edu`

Abstract

The integration of Internet of Things (IoT) technologies into computing education presents both opportunities and challenges, particularly in departments with diverse academic programs. This case study describes the development of a hardware-centric IoT course designed to serve students from computer science, cybersecurity, information systems, and information technology disciplines. The paper details the course's motivation, the adaptation of existing infrastructure, and the evaluation of hardware development kits (HDKs) based on cost, RTOS support, security, connectivity, and ecosystem compatibility. A detailed comparison of the candidate platforms is presented, leading to the selection of the ESP32-C6 for its affordability, real-time capabilities, built-in security, and support for Wi-Fi and IEEE 802.15.4. The paper also outlines laboratory exercises that reinforce core IoT concepts and offers a model for institutions aiming to implement similar cross-disciplinary IoT curricula.

1 Introduction

The interdisciplinary nature of the Internet of Things (IoT) makes it a valuable addition to any computing curriculum. All fields of computing can benefit from a study of one or more areas utilized by IoT including embedded programming, distributed computing, user experience, data management, cloud

*Copyright ©2026 by the Consortium for Computing Sciences in Colleges. Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the CCSC copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Consortium for Computing Sciences in Colleges. To copy otherwise, or to republish, requires a fee and/or specific permission.

services, product lifecycle management, attack surface awareness, and secure programming. In computing departments with diverse disciplines, however, care must be taken to design the course to address the needs of each curriculum.

Some institutions incorporate IoT concepts across multiple courses. For example, basic theory, applications, and tools may be introduced in a computer organization course, embedded system programming concepts may be covered in a programming course, and connectivity issues specific to IoT may be presented in a networking course[1]. Some institutions may provide a one- or two-semester sequence in IoT. Depending on the program of study, an IoT course may be generalized in its approach covering IoT architecture, programming, security, and networking, while other offerings may be specialized to an application such as industrial, healthcare, or agriculture[2].

Regardless of delivery method, most IoT content centers around physical hardware that interfaces with the real world and cloud-based services, suggesting that IoT content delivery relies on training the students in real-world product development to better prepare them for their careers. Burd et al. identifies hardware as a defining feature of IoT courses both for the collection of data and the control of an environment[2]. To understand the capabilities and limitations of digitally interacting with an environment, of utilizing networks for distributed devices, and of handling the security limitations of processors with limited capability, students should work with real hardware.

With respect to the hardware component of IoT, there is a wide range of options available to the students. Many microcontroller manufacturers have increased the connectivity and capabilities of their products over the past decade to keep up with the growing demands of the topic. This case study describes the selection process for an appropriate platform for a lab-based IoT course serving four disparate curricula.

2 Course Design and Adaptation

The Department of Computing at East Tennessee State University (ETSU) is in the unusual position to have four programs in one department: Computer Science (CS), Cybersecurity (CSEC), Information Systems (IS), and Information Technology (IT). This makes the design of an upper-division, cross-discipline course in IoT challenging. While all four programs are rigorous, the differences in course sequences across each are significant. Lengthy prerequisite sequences for curriculum-specific upper-division courses create problems for students who wish to take the IoT course as an elective.

The Department's IoT course was originally developed to support the CSEC curriculum with the intent to expose the students to vulnerabilities specific to IoT such as physical security weaknesses, system management issues, and increased attack surfaces[3]. By opening course enrollment to the other three

concentrations, considerations had to be made to account for the differences in student background. All four concentrations rely on a computing core consisting of 36 credit hours across twelve courses including[4]:

- Computer Science I and II
- Discrete Math
- Database Fundamentals
- Computer Organization
- Networking Fundamentals
- Information Security and Assurance
- Software Engineering I and II

Each program of study also includes courses in software and system development unique to each program. Students in the CS and CSEC programs of study are required to take a course in data structures. A course in enterprise programming is required for the IS program of study. The IT students take a course sequence covering the integration of web technologies[5]. This varied student background poses challenges for content selection and coverage in the IoT course.

As a starting point, a list of topics critical to this IoT course was developed by including the core topics strongly suggested by Burd et al. This initial list is presented below:

- IoT fundamentals (hardware)
- Computer networks (cloud/device connectivity)
- Embedded system fundamentals
- Web stacks
- Security technology and implementation
- Circuits and electronics

Topics suggested in the Burd et al. paper absent from the above list were not included due to their coverage in other courses such as a course in information and security assurance required by all four programs and a course in human-computer interaction required in the IT program and available as an elective course in the other programs.

3 Laboratory Assignment Progression

The laboratory sequence in this course was structured to incrementally introduce students to the full software–hardware stack required for modern embedded and IoT systems. Early laboratories focus on establishing a development workflow, including the installation and configuration of a modern integrated development environment and command-line toolchain, followed by foundational skills in error handling and runtime debugging. These initial exercises emphasize the need for a hardware platform with strong tooling support, robust debugging capabilities, and a stable software development framework.

Subsequent laboratories introduce real-time concepts through the use of

an embedded real-time operating system (RTOS) and interrupt service routines. Students apply these concepts to basic digital I/O and peripheral interaction, reinforcing concurrency, timing, and responsiveness—core requirements for any IoT-capable hardware platform. Peripheral interfacing is then expanded through hands-on work with serial communication protocols such as I2C. These activities require reliable peripheral drivers, flexible pin multiplexing, and sufficient memory resources.

Mid-sequence laboratories explore non-volatile storage and onboard file systems, highlighting the importance of persistent configuration data and local data storage in embedded IoT devices. This progression also provides an examination of hardware support for over-the-air (OTA) updates, secure boot, and flash encryption.

The latter portion of the course shifts toward networking and connectivity, beginning with the creation of a software-based wireless access point and evolving into more robust networked applications. Students design systems that collect credentials, transition between network roles, and expose HTTP services, emphasizing the need for reliable wireless networking, multitasking support, and protocol stack maturity. These networking concepts are further extended through the introduction of Bluetooth Low Energy (BLE), where students explore both advertising (peripheral) and scanning (central) roles, requiring hardware that supports concurrent wireless stacks and event-driven execution models.

The final laboratories integrate messaging and cloud connectivity by introducing the MQTT protocol and secure communication with a commercial IoT cloud platform. These culminating exercises require hardware support for secure networking, cryptographic services, and scalable application structure, while reinforcing the importance of end-to-end system integration.

Overall, this progression of laboratories places clear and sustained demands on the selected hardware platform: native RTOS support, comprehensive peripheral interfaces, mature wireless networking (Wi-Fi and BLE), non-volatile storage capabilities, strong development tooling, and security features suitable for cloud-connected systems. The alignment between laboratory objectives and platform capabilities was a primary consideration in the hardware selection process for the course.

4 Adapting Existing Resources

Like many institutions, the Department of Computing at ETSU has several computing laboratories set up for general-purpose software development. Some of the resources required for embedded system and IoT development could be adapted from these general-purpose setups. For example, the lab machine image contained an installation of Microsoft's Visual Studio Code. This lightweight code editor contains a powerful set of features, that when com-

bined with the vast library of extensions, allows students to effectively develop applications for IoT.

Another issue to consider was connectivity. For some activities such as configuration and testing, the students could create an ad hoc network between their IoT devices and a smartphone. To access cloud resources, however, a more robust connection was required.

Wi-Fi and Ethernet access is tightly controlled by the University's IT division to guarantee security across the campus' extensive computing infrastructure. During the development of this course, there was no support for authentication of IoT hardware with the existing campus network. Initially, this challenge was overcome using a separate Wi-Fi access point configured to use a channel that did not conflict with the University's infrastructure. A Raspberry Pi connected to this network acted as a server to which the students could connect, but this proved to be inadequate for many reasons including the inability to connect devices to commercial cloud services.

Fortunately, the IT division supported the development of this course. They created an identity pre-shared key (iPSK) network for devices not capable of WPA 3 Enterprise security, thus providing increased security. This, however, added the administrative task of gathering the media access control (MAC) addresses for all development hardware used in the lab. Once a device was registered, however, it had full access to all required cloud resources.

5 Options for Hardware Development

To address the course's hardware-oriented topics, additional development tools and equipment were deemed necessary. Specifically, students needed access to hardware development platforms and the associated development environment. Thirty years ago, engineers and programmers had to build their development platforms from components. Since then, there has been a proliferation of options for use in device development including single-board computers (SBCs), hardware development kits (HDKs), and educational platforms.

SBCs like the Raspberry Pi are fully functional computing platforms with a microprocessor, memory, and I/O interfaces integrated into a single board. The first iteration of the course utilized the Raspberry Pi 3. Students installed the Raspberry Pi OS (formerly Raspbian) and used Node.js to develop IoT projects. Node.js, with the support of Node Package Manager, provided the resources needed to achieve extensive IoT functionality.

Several issues with the platform, however, did not align with IoT development in industry. Storage, for example, is on a removeable microSD card and is more closely aligned with a desktop file system rather than the non-volatile, flat memory model of embedded systems. This feature prevented real-time operation, made the system vulnerable to malicious actors, and did not support low-power operation. Additionally, the platform did not have built-in hardware

support for encryption, low-power modes, and OTA updates. SBCs also tend to be more expensive than other hardware development options.

At the other end of the spectrum are educational platforms or kits such as the SparkFun Inventor's Kit. These kits typically come with prebuilt projects suited more for the novice with little to no computing experience and offer little to no access to the device's lower-level operation. And like the SBCs, these boards tend to be more expensive and usually provide little to no support for hardware encryption, deep sleep, and OTA updates[6].

A middle-ground solution is the hardware development kit (HDK). HDKs are small microcontroller boards that include all support necessary to allow engineers and software developers full access to the device's capabilities for prototype firmware development. Unlike educational kits, HDKs typically have little or no built-in I/O peripherals beyond an LED and a push button.

The primary benefit of HDKs over SBCs is that the HDKs match the architecture of an embedded system while the architecture of the SBC is that of a general-purpose machine. HDKs are preferred over educational kits due to lower cost and the low-level access given to the controller's features. The lower cost of the HDK allows students to purchase multiple boards to support multiple projects, inter-device communication, and backups in case of hardware failure. It is for these reasons that the decision was made for the spring semester of 2022 to shift from the Raspberry Pi 3 to an HDK. The remainder of this paper covers the process used to select the HDK on which to base the course.

6 Selection Criteria

Selecting a suitable microcontroller for a new project can be daunting in both industry and academia. In industry, designers often rank project requirements and apply tools such as weighted rubrics to evaluate options. A similar process applies when choosing a processor architecture for an IoT course, though the weighting of criteria differs. For example, hardware cost may be secondary in industry but becomes critical when students or departments must purchase devices in large quantities. Conversely, while development ecosystem fit strongly impacts time-to-market in industry, academia may prioritize exposing students to a variety of ecosystems.

For this course, the development of a list of decision criteria began with the topic list presented earlier in this paper. This list was then updated based on topics found consistently in IoT courses across respected universities. Lastly, faculty from all four programs in the Department were encouraged to add criteria they felt were important to the course. The faculty then were asked to sort the criteria in order of importance. The result of this process is presented below as a list in order of importance beginning with the highest. One criterion, Wi-Fi or IEEE 802.15.4 capabilities, was left off the list as it was considered

nonnegotiable.

1. Peripheral interfacing (wired and wireless protocols)
2. Cost and availability for students purchasing hardware
3. Toolchain and ecosystem compatibility with other courses
4. Real-time operation
5. Hardware-based encryption and secure boot
6. Over-the-air (OTA) update capabilities
7. Cloud integration support
8. Power management
9. Data privacy
10. Processor performance

The following section presents a list of HDKs considered for use in the course.

7 HDKs Considered

With the decision made to go with HDKs, a list of Wi-Fi or IEEE 802.15.4 capable candidate platforms was created using the top three requirements identified in the previous section: digital peripheral support such as I2C, SPI, and BLE, low-cost to the student, and a well-supported development ecosystem. Table 1 presents a condensed list of the platforms identified and some of their basic characteristics. Many HDKs were evaluated that are not included in this list. For example, several Espressif HDKs were candidates, but the ESP32-C6 was selected to represent the platform due to its processor architecture.

Table 1: Core SDK Capabilities Comparison

Platform	Cost	Flash / SRAM	IDE Ecosystem
Arduino Nano 33 IoT[7]	\$28	256 kB / 32 kB	Arduino, VS Code, PlatformIO
Arduino UNO R4 WiFi[8]	\$28	256 kB / 32 kB	Arduino, VS Code, PlatformIO
ESP32-C6[9]	\$10	16 MB / 512 kB	Arduino, VS Code, ESP-IDF, PlatformIO
nRF52840 Dongle[10]	\$10	1 MB / 256 kB	nRF Connect SDK, VS Code, Segger
NUCLEO-WB55RG[11]	\$36	1 MB / 256 kB	STM32CubeIDE, VS Code, Keil, IAR, Arduino, PlatformIO
Raspberry Pi Pico W[12]	\$12	4 MB / 520 kB	Thonny, VS Code

8 Platform Comparison

All selected HDKs support digital and analog I/O, but only the Arduino UNO R4 WiFi, ESP32-C6, Nordic nRF52840, and NUCLEO boards provide built-in user-programmable buttons and one or more LEDs. The nRF52840 MDK USB Dongle and ESP32-C6 HDKs are particularly capable in that they provide RGB LEDs for more robust output. The built-in output of the UNO R4 WiFi goes above and beyond with an LED matrix. Except for minor differences such as the Arduino UNO R4 WiFi and the ESP32-C6 supporting capacitive touch sensing and CAN bus and the Nordic nRF52840 providing NFC with an additional antenna, all the HDKs support multiple board-level protocols including I2C, SPI, I2S, and UART, sufficient to interface to intelligent sensors.

As discussed earlier, one of the course's learning outcomes focuses on an understanding of IoT security. All of the platforms except the Raspberry Pi Pico W provide integrated hardware-based security such as flash encryption and secure boot. Unfortunately, aside from a hardware-based PRNG, security support for the Raspberry Pi Pico W is software-based. Three of the platforms, ESP32-C6, nRF52840 Dongle, and NUCLEO-WB55RG, provide excellent built-in security support with features such as WPA3 in the ESP32-C6 and ARM TrustZone and crypto acceleration in the Nordic nRF52840 MDK USB Dongle and the NUCLEO-WB55RG.

Although all of the platforms claim to support RTOS, only three of them (Espressif ESP32-C6, Nordic nRF52840 MDK USB Dongle, and NUCLEO-WB55RG) support full-featured RTOS operation. The Arduino platforms are largely abstracted while the Raspberry Pi Pico W requires additional manual setup/configuration to become operational.

While all of the platforms allow OTA updates, only three of them (Espressif ESP32-C6, Nordic nRF52840 MDK USB Dongle, and NUCLEO-WB55RG) integrate multiple firmware slots allowing a new firmware image to be downloaded to a secondary, inactive flash memory slot while the current application is still running from the primary slot. The other platforms require software-based OTA support to handle updates.

All but the Arduino UNO R4 WiFi HDK use 3.3 volt digital I/O to support low power operation, so a bi-directional logic level converter will most likely be required. This is because many peripherals operate on 5 volts meaning that an HDK operating on 3.3 volts would require a bi-directional logic level converter to be incorporated into circuitry. These devices would add approximately 5 to 10 to the cost. As far as low power operation goes, there are slight differences in built-in support with the Raspberry Pi Pico W having the most limited of the evaluated boards.

All the evaluated HDKs have Microsoft Visual Studio Code extension support, but more weight was given to devices with compilers compliant to a

language standard such as C/C++. Although other open-source RTOSes are gaining popularity, more weight was also given to devices that natively support FreeRTOS.

While not part of the original decision criteria, each platform possessed features making it more or less attractive. For example, the Arduino platforms provide additional cloud support via Arduino IoT Cloud, yet they also utilized a non-standard implementation of C/C++. The Espressif ESP32-C6 is based on the open-source RISC-V architecture, an architecture used in some of the other courses at this institution. The Nordic nRF52840 MDK USB Dongle and NUCLEO-WB55RG platforms have full IEEE 802.15.4 support allowing for Zigbee, Thread, and 6LoWPAN applications, yet they require expansion boards/shields to support Wi-Fi. The Raspberry Pi Pico W had support for a debug probe when connected to a Raspberry Pi 4/5.

Due to its low cost, robust C/C++ support through the VS Code ESP-IDF extension, native FreeRTOS integration, wide range of low-power operating modes, and strong hardware support for secure boot, flash encryption, and OTA updates, the ESP32-C6 was selected as the HDK for this course. An additional advantage of the ESP32-C6 is its open-source RISC-V architecture, which is also employed in two other courses within the department, enabling architectural continuity across the curriculum.

9 Derived Course Materials

The following is a summary of some of the labs developed and taught for multiple semesters using the ESP32-C6 platform. Not included in the list are some of the basic labs such as an introduction to the platform and error handling in embedded system design.

- Task and interrupt handling in FreeRTOS
- I2C and SPI interfacing and protocols
- Embedded system memory technology best practices
- SW-based Wi-Fi AP and server implementation for device configuration
- Secure BLE connections to embedded device
- MQTT publish and subscribe operation
- Creation of IoT device on a cloud service
- Data collection and management on a cloud service

Each of the labs was performed during a two-hour lab period on general purpose computers running Microsoft Visual Studio Code with the ESP-IDF extension installed. When connectivity was required, the iPSK network was utilized.

10 Conclusion

This case study demonstrates a workable model for selecting and implementing hardware in a cross-disciplinary IoT course. For this course, the ESP32-C6

HDK was selected as it provides students with a practical platform to explore real-world embedded system and IoT design. The structured evaluation approach used ensured alignment with the course’s pedagogical and technical goals.

References

- [1] Ahmed Abdelmoamen Ahmed et al. “Integrating IoT Technologies into the CS Curriculum at PVAMU: A Case Study”. In: *Education Sciences* 12.11 (2022), p. 840. URL: <https://doi.org/10.3390/educsci12110840> (visited on 07/10/2025).
- [2] Barry Burd et al. “Courses, Content, and Tools for Internet of Things in Computer Science Education”. In: *Proceedings of the 2017 ITiCSE Conference on Working Group Reports (ITiCSEWGR '17)*. New York, NY, USA: ACM, 2018, pp. 125–139.
- [3] East Tennessee State University Department of Computing. *CSCI 4677/5677 - Internet of Things - New Course Proposal (internal document)*. 2018.
- [4] Tony Pittarese et al. “Developing an Integrated Multi-Program Computing Department”. In: *Journal of Computing Sciences in Colleges* 30.5 (2015), pp. 20–26. URL: <https://dl.acm.org/doi/10.5555/2752981.2752985> (visited on 05/06/2024).
- [5] East Tennessee State University. *2025-2026 Undergraduate Catalog*. 2024. URL: <https://catalog.etsu.edu/index.php?catoid=61> (visited on 07/10/2025).
- [6] Biju Bajracharya, Vamsi Gondi, and David Hua. “IoT Education using Learning Kits of IoT Devices”. In: *Information Systems Education Journal* 19.6 (2021), pp. 40–44. URL: <https://files.eric.ed.gov/fulltext/EJ1329491.pdf> (visited on 07/17/2025).
- [7] Arduino S.r.l. *Arduino Nano 33 IoT*. Arduino Official Store. n.d. URL: <https://store-usa.arduino.cc/products/arduino-nano-33-iot> (visited on 07/17/2025).
- [8] Arduino S.r.l. *Arduino UNO R4 Wi-Fi*. Arduino Official Store. n.d. URL: <https://store-usa.arduino.cc/collections/uno/products/uno-r4-wifi> (visited on 07/17/2025).
- [9] Espressif Systems. *ESP32-C6 series datasheet (Version 1.3)*. 2023. URL: https://www.espressif.com/sites/default/files/documentation/esp32-c6_datasheet_en.pdf (visited on 07/17/2025).
- [10] Makerdiary. *Hardware description - nRF52840 MDK USB Dongle*. n.d. URL: <https://wiki.makerdiary.com/nrf52840-mdk-usb-dongle/hardware/> (visited on 07/17/2025).
- [11] STMicroelectronics. *NUCLEO-WB55RG*. n.d. URL: <https://www.st.com/en/evaluation-tools/nucleo-wb55rg.html> (visited on 07/17/2025).
- [12] Raspberry Pi Foundation. *Raspberry Pi Pico*. n.d. URL: <https://www.raspberrypi.com/products/raspberry-pi-pico/> (visited on 07/17/2025).

From Points to Proficiency: How Students Perceive Standards-Based Grading in Computing Courses*

Jonathan McCurdy and Ruth Lamprecht
Mathematics and Computer Science
Mount St. Mary's University
Emmitsburg, MD 21727
`{j.r.mccurdy,r.e.lamprecht}@msmary.edu`

Abstract

Standards-based grading (SBG) emphasizes proficiency on clearly defined learning goals through feedback and reassessment rather than point accumulation. This study reports results from a survey of 90 students in computing courses examining their perceptions of SBG as a learning-focused and rigorous grading practice. Students responded positively to reassessment opportunities, clear expectations, and actionable feedback, and many reported that SBG supported persistence, reduced stress, and encouraged mastery-oriented learning. Responses were more mixed on grade interpretation, as some expressed discomfort with binary proficiency grading and uncertainty about their standing in the course. These findings suggest that SBG can promote positive learning experiences in computing courses while highlighting the importance of transparency and progress tracking for sustaining student buy-in.

1 Introduction

Standards-based grading (SBG) is an assessment philosophy that shifts the focus from accumulating points to measuring student proficiency on clearly defined learning goals [1, 2, 3]. In contrast to traditional methods that often aggregate a “hodgepodge” of academic performance and behavioral factors, SBG

*Copyright ©2026 by the Consortium for Computing Sciences in Colleges. Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the CCSC copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Consortium for Computing Sciences in Colleges. To copy otherwise, or to republish, requires a fee and/or specific permission.

isolates and evaluates a student's terminal mastery of specific content standards [3]. As outlined by Beatty [4] (citing Marzano, Dueck, and Brookhart), this philosophy is anchored by three essential pillars: feedback should align with clearly defined learning objectives to guide students toward what they still need to master; students should have opportunities to address these gaps through reassessment; and grades ought to reflect mastery of content rather than non-cognitive factors.

The theoretical foundation of mastery-oriented learning is rooted in Bloom's Mastery Learning and Keller's Personalized System of Instruction, both of which emphasize that learning outcomes are maximized when students receive formative feedback, sufficient time, and repeated opportunities for demonstration of competence. Recent implementations in higher education have extended this philosophy through a variety of models—SBG, mastery-based testing, and specifications grading—each emphasizing clarity, feedback, and opportunities for growth rather than cumulative point collection [3, 5, 6].

2 Related Work

Mastery-based systems are often justified both as fairer representations of achievement and as instructional designs that shape how students learn throughout the semester. By making proficiency the target and allowing repeat attempts, these systems can shift assessment from a one-time judgment to an ongoing feedback process that shapes effort and persistence. Accordingly, higher education research often examines three interconnected outcomes: student learning, motivation, and stress.

2.1 Impacts on Student Learning, Motivation, and Stress

Learning: Prior work links mastery-oriented assessment to stronger conceptual learning and course performance. Campbell et al. [3] described mastery grading systems in which eventual mastery, rather than first-attempt performance, determines the final grade, allowing learning to unfold iteratively throughout the semester. In controlled comparisons, mastery-based testing has predicted final exam performance more accurately than point-based quizzes [5].

Motivation: SBG and related systems may also enhance motivation by aligning with the principles of self-determination theory: competence, autonomy, and relatedness [7]. Students in mastery-graded courses report higher intrinsic goal orientation and self-efficacy than peers in traditional sections [8]. Tripp et al. [7] found that in-class reassessment opportunities increased students' sense of control over outcomes, turning exams into opportunities for deliberate practice rather than threats to self-esteem.

Stress and Anxiety: Another recurring finding is reduced test anxiety. Students often describe mastery-based courses as less punitive because grades depend on demonstrated proficiency rather than the grade depending on if they made mistakes early in the course. Harsy [6] found that students in a

mastery-based calculus sequence reported lower pre-exam stress and greater perceived fairness than those in a traditionally graded cohort. Lewis et al. [9] similarly documented decreases in self-reported grade anxiety when multiple reassessment opportunities were integrated into mathematics courses.

2.2 Implementation and Challenges

Despite their benefits, SBG systems require substantial preparation and refinement. Instructors must articulate measurable learning objectives, construct multi-version assessments, and develop tracking systems for reassessment data [10, 2]. Faculty reports highlight two persistent challenges: initial workload and student acclimation. Instructors often describe significant time costs during the first implementation cycle [11], while students unfamiliar with SBG may initially perceive binary “proficient/not-yet” scoring as unforgiving [12]. However, perceptions often improve after the first reassessment cycle, as students recognize that mastery grading allows multiple pathways to success and values persistence over perfection [13]. Best practices recommend limiting the number of standards, using concise proficiency scales, and mapping graded tasks directly to learning outcomes [10].

2.3 Focus on Computer Science and Technical Courses

In computing and other technical disciplines, mastery learning aligns well with iterative skill development and debugging practices [14]. Mastery-oriented structures can ease the early learning curve of syntax and abstraction, while modified specifications grading lets students revise code until it meets specified criteria [15]. A longitudinal study at the University of Otago found that students who completed a mastery-based Python course performed better in the subsequent Java course, with the largest gains among lower-performing students [16]. However, mastery learning’s flexibility can also encourage procrastination. Ott et al. [17] observed that students in self-paced mastery models often delayed reassessment until late in the semester, creating grading bottlenecks and uneven engagement. Structured scheduling, such as retake limits or token systems, has been proposed as a remedy [11].

Recent computing education reviews also help clarify what “standards” often look like in CS implementations. Cheng et al.’s [18] review found that introductory computing assessments most often rely on code-writing and multiple-choice formats, balancing authentic programming competence with grading feasibility at scale. This matters for SBG because the approach is most actionable when standards are concrete, assessable, and repeatedly sampled across the term.

2.4 Overview

Collectively, the literature suggests that SBG-like systems can promote more accurate representations of learning, greater motivation, and reduced anxiety, particularly in STEM and computing contexts. Their effectiveness depends

heavily on implementation details such as clarity of objectives, pacing structures, and feedback mechanisms. While many studies document these benefits, relatively few focus specifically on student perceptions of SBG in computing-related courses. Our study contributes to this work by examining how students in computing courses perceive SBG, including its effects on clarity, persistence, stress, and academic rigor.

3 Findings of Study

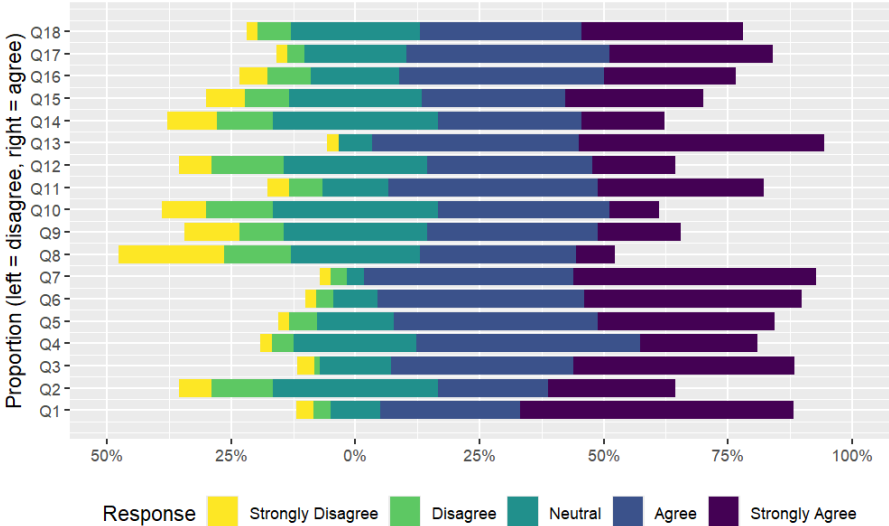
For this study, we surveyed 90 students across multiple courses utilizing SBG within the department to gauge students’ perceptions of the practice. These courses included introductory programming and data science courses, Machine Learning, Data Structures & Algorithms, Computer Architecture, and Computer Security. Of the 90 students who participated, 68 (75.6%) declared a technology-related major (computer science, cybersecurity, or data science). Additionally, 58 (64.4%) indicated that they had taken a class before which utilized standards-based grading.

Post-semester responses were summarized descriptively as the proportion of students selecting each Likert option (Strongly Disagree through Strongly Agree). Because the purpose of this study was to characterize student perceptions rather than test a causal intervention effect, we report top-2 box proportions (Agree + Strongly Agree) together with 95% confidence intervals to summarize the strength and uncertainty of endorsement for each item without imposing interval-scale assumptions on Likert responses. If the entire 95% CI for the proportion is above 50%, we interpret that item as showing clear majority agreement within the sampled population (see Fig. 1). While the questions were not grouped in the survey, Table 1 organizes them by analytic purpose. To contextualize these descriptive quantitative patterns, we integrate representative excerpts from the free-response questions throughout the analysis.

3.1 Pertaining to Feedback and Clear Goals

Feedback and clear goals were consistently endorsed by surveyed students. A majority agreed that *I think having clear expectations for each standard helped reduce my uncertainty about how to succeed* (Q3) and that *Being assessed on specific learning goals helped me understand what I need to work on* (Q6). Students’ qualitative comments aligned closely with these perceptions of transparency and actionable guidance, with one student saying that it was beneficial to know, “...where I had to be or work up to”. Several students also emphasized that standards created diagnostic insight into their learning, noting that the reassessments “allowed me to get a real-time understanding about where I struggle” and that “knowing which part of the standard I got wrong and why helped because it made sure I actually knew what was being asked.”

Post-Survey Likert Response Distributions (Centered at Neutral)



95% CI of Proportion Agreeing (Agree or Strongly Agree)

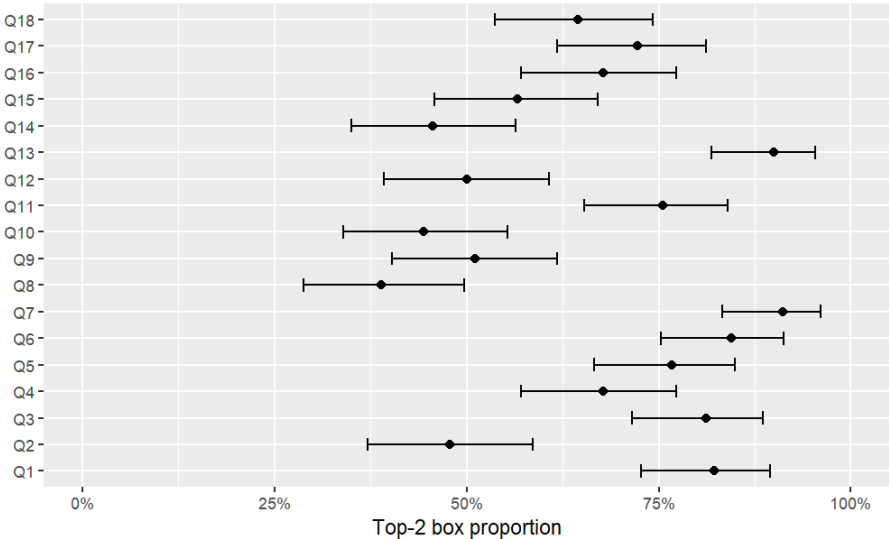


Figure 1: Item Endorsement Barplot and Agreement Confidence Intervals

Table 1: Post-survey item endorsement (top-2 box proportions with 95% CI)

Item	Statement	Agree/SA %	[95% CI]
<u>Feedback / Clear Goals</u>			
Q3*	I think having clear expectations for each standard helped reduce my uncertainty about how to succeed	81	[72, 89]
Q5*	I paid about the same attention to feedback as I usually do in graded classes	77	[67, 85]
Q6*	Being assessed on specific learning goals helped me understand what I need to work on	85	[76, 92]
<u>Stress and Anxiety</u>			
Q1*	The ability to revise or reattempt assignments reduced my academic stress	83	[74, 90]
Q8	I felt less anxious about grades in this course compared to others	39	[29, 50]
Q11*	Knowing that I didn't have to be perfect on the first try helped reduce my anxiety about assessments	76	[65, 84]
<u>Motivation / Growth Mindset</u>			
Q9	A standards-based system helped me focus better on learning	51	[40, 62]
Q13*	My ability in this subject can improve through effort	91	[83, 96]
Q14	SBG increased my intrinsic motivation to learn and understand course material	46	[35, 56]
Q16*	The grading system helped me focus on learning rather than just earning points	68	[57, 77]
<u>Persistence</u>			
Q4*	I believe SBG motivated me to improve even after getting things wrong	69	[58, 78]
Q7*	Having the opportunity to be reassessed encouraged me to keep trying, even after setbacks	91	[83, 96]
<u>Academic integrity / AI</u>			
Q15	Knowing I could resubmit work made me less hesitant to experiment with AI tools	57	[46, 67]
Q17*	I believe standards assessment supports academic honesty	74	[63, 83]
Q18*	Allowance for resubmitting my work reduced my interest in using AI	65	[54, 75]
<u>Grade Meaning</u>			
Q2	I feel comfortable not receiving traditional letter grades on individual assignments	48	[37, 59]
Q10	My final grade reflected what I actually understood because of the grading system	44	[34, 55]
Q12	I felt less pressure to compete with classmates when grading was based on individual standards	50	[39, 61]

Notes: Agree/SA % is the top-2 box proportion (Agree + Strongly Agree). 95% CIs are exact binomial intervals. An asterisk (*) indicates the CI is entirely above 50%, consistent with majority agreement. Item-level n was approximately 90.

3.2 Pertaining to Persistence

Items reflecting persistence after making mistakes showed some of the strongest endorsements. Students connected with the statements that *I believe SBG motivated me to improve even after getting things wrong* (Q4) and, even more strongly, that *Having the opportunity to be reassessed encouraged me to keep trying, even after setbacks* (Q7). This suggests that reassessment was perceived as a meaningful support for continued effort after unsuccessful initial attempts. Open-ended responses reinforced this interpretation, with students saying that “it made me motivated to improve because the failure was not permanent” and “it would make me more motivated to try even if I’m wrong and to keep trying till I’m right”. Others explicitly connected reassessment to mastery-oriented engagement with course content, saying “retaking standards will help my overall learning experience because I can take the time to realize what I’m doing wrong and what to work on” and “it encouraged me to learn the material instead of cram information”.

3.3 Pertaining to Motivation and Growth Mindset

Motivation and learning orientation were generally positive, but not uniform across items. Students overwhelmingly endorsed the growth-mindset statement *My ability in this subject can improve through effort* (Q13), and a majority also agreed that *The grading system helped me focus on learning rather than just earning points* (Q16). Students mentioned an emphasis on improvement and learning, saying “it made me look at the work as improvement instead of just a grade” and “this will help improve a growth mindset by encouraging students to keep trying even after failing the first time”. Students also described their increased willingness to engage with the material because “getting to re-submit would make me more eager to take the feedback and try and learn...”. As seen in other literature, some students occasionally reported less motivation knowing they had a second chance, stating that “reassessment increased motivation...but it sometimes made me procrastinate”. Responses were divided for *A standards-based system helped me focus better on learning* (Q9) and for *SBG increased my intrinsic motivation to learn and understand course material* (Q14). Qualitative comments suggest that, for some students, the motivational benefits were tempered by the fatigue of reassessments, saying “it was good initially, but at some point after around 5 or so retakes it gets tiring”.

3.4 Pertaining to Stress and Anxiety

Questions relating to stress and anxiety were more nuanced, showing a separation of assessment-related stress from broader grade anxiety. Most of the students agreed that *The ability to revise or reattempt assignments reduced my academic stress* (Q1) and that *Knowing that I didn’t have to be perfect on the first try helped reduce my anxiety about assessments* (Q11), consistent with the idea that reassessment without penalty reduces the stakes of any single per-

formance. Qualitative responses echoed this reduced-pressure framing, with a student characterizing reassessment as a safety net by saying “it gives me the ability to retry, so it removes perfectionism and a lot of stress”. In contrast, the endorsement was weaker for *I felt less anxious about grades in this course compared to others* (Q8), where responses were mixed and did not have a clear majority in support. The combined findings suggest that reassessment opportunities may reduce stress tied to individual assignments, while grade anxiety may persist for the overall course grade.

3.5 Pertaining to Academic Integrity and AI Usage

Students tended to agree that *I believe standards assessment supports academic honesty* (Q17) and that *Allowance for resubmitting my work reduced my interest in using AI* (Q18), suggesting that students viewed the system as supporting honest work and reducing the perceived need to use AI as a shortcut. This theme surfaced in the free-response questions, with one student stating that “it would motivate me to really try MY best instead of asking for help from people or AI”. At the same time, *Knowing I could resubmit work made me less hesitant to experiment with AI tools* (Q15) showed mixed endorsement. This might suggest that AI use may be interpreted by the students in more than one way; a tool for learning and exploration, and a tool to avoid learning.

3.6 Pertaining to Grade Meaning

Finally, grading-norm items were among the least strongly endorsed, and had comparatively high neutral response rates. Students were mixed on *My final grade reflected what I actually understood because of the grading system* (Q10) and *I feel comfortable not receiving traditional letter grades on individual assignments* (Q2), and *I felt less pressure to compete with classmates when grading was based on individual standards* (Q12). The qualitative data gives some insight into these mixed endorsements, with some students expressing discomfort and frustration with the nontraditional grading structures and the harshness of all-or-nothing scoring. They said “I did not like the concept of not giving partial credit” and “needing to have a perfect score to pass or else 0”. Others thought that it better showed their learning progress by stating that it is “...better since I feel like my grades are more accurate”. Overall, these results suggest that while students clearly recognize the benefits related to persistence and reduced assessment-level stress, more can be done to promote student buy-in for the non-traditional grading approach.

4 Discussion

Across the courses in this study, students generally described SBG as making expectations clearer, lowering the stakes of any single attempt, and encouraging persistence through reassessment. The strongest endorsement appeared in items related to reassessment, feedback, and clear learning goals, while re-

sponses were more mixed for grade interpretation, intrinsic motivation, and overall course-grade anxiety. These patterns are consistent with prior work on mastery-oriented assessment, where clear objectives and opportunities to improve are associated with stronger learning orientation and lower assessment-related stress.

At the same time, student comments suggest that the benefits of SBG depend heavily on implementation. While many students viewed the system as fairer and more growth-oriented, others expressed frustration with binary scoring, uncertainty about their standing in the course, and reassessment backlog. SBG must also be designed carefully so that standards and reassessments do not overemphasize only the most easily measurable skills at the expense of more nuanced understanding. In practice, effective SBG in technical courses appears to require clear expectations, simple progress tracking, and pacing structures that support continued engagement without overwhelming students.

These findings should be interpreted with several limitations in mind. This study captures self-reported perceptions rather than direct measures of learning, comes from a single department at one institution, and does not support causal claims about the effects of SBG relative to traditional grading. Even so, the results suggest that SBG can support positive learning experiences in computing courses when paired with transparent course design and clear communication about progress and grading. Future work should examine how these perceptions relate to learning outcomes and which implementation choices most improve student buy-in.

References

- [1] Valerie L Marsh. *Standards-Based Grading: History, Practices, Benefits, Challenges, and Next Steps*. Tech. rep. University of Rochester's Warner School of Education, 2023.
- [2] William J Howitz, Kate J McKnelly, and Renée D Link. "Delving into the Design and Implementation of Specifications Grading Systems in Higher Education". In: *Education Sciences* 15.1 (2025), p. 83.
- [3] Robert Campbell, David Clark, and Jessica OShaughnessy. *Introduction to the special issue on implementing mastery grading in the undergraduate mathematics classroom*. 2020.
- [4] Ian D Beatty. "Standards-based grading in introductory university physics". In: *Journal of the Scholarship of Teaching and Learning* (2013), pp. 1–22.
- [5] J B. Collins et al. "Mastery-based testing in undergraduate mathematics courses". In: *Primus* 29.5 (2019), pp. 441–460.
- [6] Amanda Harsy and Alyssa Hoofnagle. "Comparing mastery-based testing with traditional testing in calculus II". In: *International Journal for the Scholarship of Teaching and Learning* 14.2 (2020), p. 10.

- [7] Brie Tripp et al. “Undergraduate STEM Students’ Perceptions of Grading Practices Reveal that Quiz Retakes Positively Impact Drivers of Self-determination”. In: *CBE—Life Sciences Education* 24.1 (2025), ar4.
- [8] Melanie Butler. “Alternative grading systems and student outcomes: a comparative analysis of motivation, enjoyment, engagement, stress, and perceptions of final grades”. In: *Assessment & Evaluation in Higher Education* (2025), pp. 1–11.
- [9] Drew Lewis. “Impacts of Standards-Based Grading on Students’ Mindset and Test Anxiety.” In: *Journal of the Scholarship of Teaching and Learning* 22.2 (2022), pp. 67–77.
- [10] Adam R Carberry et al. “Best practices for using standards-based grading in engineering courses”. In: *2016 ASEE Annual Conference & Exposition*. 2016.
- [11] Ruth Lamprecht et al. “Standards-Based Grading in Undergraduate Courses for Technology Majors”. In: *6th International Computer Programming Education Conference (ICPEC 2025)*. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. 2025, 10:1–10:14.
- [12] Kate J McNelly et al. “Specifications grading at scale: Improved letter grades and grading-related interactions in a course with over 1,000 students”. In: *Journal of Chemical Education* 100.9 (2023), pp. 3179–3193.
- [13] Tom Buckmiller, Randal Peters, and Jerrid Kruse. “Questioning points and percentages: Standards-based grading (SBG) in higher education”. In: *College Teaching* 65.4 (2017), pp. 151–157.
- [14] Lijun Chen et al. “Experience report: Standards-based grading at scale in algorithms”. In: *Proceedings of the 27th ACM Conference on on Innovation and Technology in Computer Science Education Vol. 1*. 2022, pp. 221–227.
- [15] Kevin R Sanft, Brian Drawert, and Adam Whitley. “Modified specifications grading in computer science: Preliminary assessment and experience across five undergraduate courses”. In: *Journal of Computing Sciences in Colleges* 36.5 (2021), pp. 34–46.
- [16] Brendan McCane et al. “Mastery learning in introductory programming”. In: *Proceedings of the nineteenth australasian computing education conference*. 2017, pp. 1–10.
- [17] Claudia Ott, Brendan McCane, and Nick Meek. “Mastery learning in cs1—an invitation to procrastinate?: Reflecting on six years of mastery learning”. In: *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1*. 2021, pp. 18–24.
- [18] Qingwan Cheng et al. “Design an assessment for an introductory computer science course: A systematic literature review”. In: *2022 IEEE frontiers in education conference (FIE)*. IEEE. 2022, pp. 1–8.

Scaffolded and Culturally Relevant Python Learning with AI as a Socratic Partner in K–12 Computing*

Marlon Mejias¹, Oluwagbeminiyi Johnson¹, Abi Olukeye¹,
Luce-Melissa Kouaho¹, Victor Mack²

¹Department of Software and Information Systems

²Center for Science, Technology, Engineering
& Mathematics (STEM) Education

University of North Carolina at Charlotte
Charlotte, NC 28223

mmejias@charlotte.edu, ojohns16@uncc.edu

{aolukeye, lkouaho, vbmack}@charlotte.edu

Abstract

We present a breadth-first, scaffolded K–12 computing model that integrates block-based coding, Python, machine learning, physical computing, and generative AI within a culturally relevant, constructivist framework. A central innovation is the use of ChatGPT as a Socratic partner: students are taught a structured prompt routine that elicits questions, partial hints, and small checks of understanding, thereby supporting just-in-time (JIT) learning and metacognition while avoiding answer-giving. In a two-week camp with 32 students, we observed consistent patterns across middle and high school cohorts: high engagement, a manageable transition from blocks to text when anchored to a single task (Rock–Paper–Scissors), and strong perceived value in AI constrained to hints. No statistically significant differences emerged between cohorts; both reported gains in confidence, strategy use, and computational thinking. We outline practical routines (e.g., anchoring across modalities, JIT

*Copyright ©2026 by the Consortium for Computing Sciences in Colleges. Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the CCSC copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Consortium for Computing Sciences in Colleges. To copy otherwise, or to republish, requires a fee and/or specific permission.

concept timing, lightweight reflection artifacts) and guardrails (e.g., capability framing for agency and equity). This approach is feasible for classrooms seeking to integrate Python, introductory ML, and physical computing without overwhelming novice learners.

1 Introduction

In computing education, introductory experiences often shape whether students see themselves as capable and interested in continuing the computing journey. For students in K–12, early exposure to computer science must go beyond syntax and logic to also build identity, agency, and relevance. This paper presents a two-week summer camp that leveraged a constructivist, culturally relevant, and JIT learning framework to cultivate foundational computing skills, spark curiosity, and strengthen student motivation through a multimodal curriculum integrating block-based coding, Python, machine learning, cybersecurity, and artificial intelligence.

This paper builds on our previous work [1] and contributes a replicable data-informed framework for the education of computer science assisted by Artificial Intelligence (AI). New contributions include: (i) ten research questions aligned with evaluation metrics; (ii) a *socratic* prompt engineering framework for Generative AI; (iii) mixed-methods evaluation including statistical analysis and thematic coding; (iv) a refined curricular scaffold progressing from block-based coding to text-based AI-integrated projects.

2 Background and Related Work

K–12 computing research highlights the importance of *active making*, *identity-affirming tasks*, and *timely scaffolds*. Constructivist and inquiry-based environments position learners as builders who iteratively test and revise ideas [2, 3], while just-in-time (JIT) instruction provides brief, well-timed explanations that sustain momentum and metacognition [4]. Microlearning research similarly emphasizes the value of small, targeted interventions [5]. Culturally relevant computing extends these principles by connecting computing concepts to students’ lived experiences and community contexts, which supports motivation and persistence [6, 7, 8].

Our prior work applied these ideas in a week-long camp organized around a single anchor problem revisited across multiple programming modalities [1]. Holding the problem constant while changing representational form helped reduce cognitive load, supported JIT micro-scaffolds, and provided culturally responsive entry points for diverse learners.

A central construct in this work is *computational agency*, students’ ability to make independent, well-justified choices with computing tools [9, 10]. Research highlights the challenges of transitioning from block-based to text-based programming and suggests the need for scaffolded approaches[11]. In this work, we bridge this shift by holding the underlying problem constant while increas-

ing complexity. As described in section 3, our anchor task serves this bridging function.

Generative AI introduces new opportunities and risks for novice agency. Large language models (LLMs) can act as dialogic supports that surface assumptions, prompt reflection, and provide timely checks when used with clear roles and norms [12, 13]. To avoid answer-giving and dependence, we adopt the *capability approach* [14], which frames AI as valuable when it broadens learners’ real options for planning, testing, and justifying solutions while keeping control of reasoning with the student.

This iteration extends our prior work by (i) formalizing a Socratic prompting routine that operationalizes the capability lens, (ii) refining the single-anchor scaffold that bridges blocks → Python → ML → devices, and (iii) reporting mixed-methods evidence from a new cohort.

3 Camp Design and Implementation

We used a culturally relevant, JIT, breadth-first sequence that moved from authentic discussion and no-code sense-making to code-along and learner report-outs each day (recap → discussion → no-code → code-along → report-out). This contrasts with traditional depth-first instruction, which often delays engagement with real-world applications until after foundational topics are mastered. Instead of front-loading syntax and isolated skills, our model prioritized early exposure to meaningful tasks, followed by layered revisits as complexity grew.

This one-week camp scaffolded novices, in two separate cohorts (17 middle school students and 15 high school students), from introductory coding to applications in data science, HCI, AI/ML, Physical Computing, and Cybersecurity. Our instruction emphasized computational thinking (the process of problem solving using decomposition, pattern recognition, abstraction, and algorithmic design), culturally relevant tasks, student agency, and brief reflections. Facilitators included a faculty lead, a graduate assistant, both of whom had prior training in culturally responsive and inclusive pedagogy, and an undergraduate mentor.

3.1 Curricular Progression

We used a single anchor, Rock–Paper–Scissors (RPS), to bridge blocks to Python, and then to ML input and Micro:Bits. Concepts were introduced *just in time* and tied to the anchor to reduce split attention and reinforce computational thinking as a problem-solving habit [15].

Students began in Blockly (Plane Seat Calculator, and Maze) to practice sequence, loops, and conditionals; We then mapped those ideas explicitly to Python syntax and structure to ease the block → text transition [11]. Short try–check–revise cycles and ``concept + context`` searches (e.g., `random` for computer movement) kept momentum while modeling metacognitive strategies.

Table 1: Constructivist and AI-augmented learning progression

Topic Area	Activity or Project	Key Learning Concepts
Computational Thinking (CT)	Blockly Plane Seat Calculator & Maze	Variables; Sequence, Loops, Conditionals; Debugging
Algorithmic Thinking	Rock–Paper–Scissors algorithms	CT (decomposition, pattern recognition, algorithm design)
Core Project: RPS (linear)	Build RPS with conditionals	Variables/abstraction; Logical operators & Branching; Libraries; Debugging
RPS Refactor + AI Support	Modularize game; Use ChatGPT for guidance	Abstraction; Function design; Prompt Engineering
Machine Learning integration	Teachable Machine gesture input for RPS	Classification; Model Integration; User-input Redesign; Packages
Physical computing & Cyber-security	Two-player RPS on Micro:Bit radios	Communication Protocols; Message handling; Error Management

Students also completed a data-science activity (e.g., visualizing cohort demographic data (MS), iPhone price prediction modeling HS) and wrote brief daily reflections. For detailed learning outcomes and daily structure, including discussion formats, code-along routines, and learner report-outs, see Section 3 of [1].

3.2 Ethical and Pedagogical Role of Generative AI

We treated generative AI as a *partner in cognition*, not an answer engine [14]. To keep students responsible for the work of thinking, our prompts: (i) invite clarifying questions and partial hints (not full solutions), (ii) request small, stepwise checks of understanding, and (iii) require brief reflection on errors, alternatives, and next steps. This structure avoids deskilling and builds a durable understanding that students can carry into new problems [14]. In practice, the interaction stays lightweight: learners ask for the smallest useful next step, implement it themselves, and then verify.

3.2.1 Prompt Engineering and Effective AI Interaction

Rather than relying on trial and error, we explicitly taught students how to structure prompts that encourage the AI to function as a supportive learning partner. The components of a good prompt include:

- **Instruction** – clearly state what the AI should do;
- **Context** – provide background about the task or problem;
- **Input data** – describe what the user is working with or struggling with;
- **Output indicator** – clarify the desired format or kind of response;
- **Constraints** – add any limits or boundaries (e.g., “don’t give the full answer”).

Example prompt used with RPS:

“You are going to act as a Python tutor. I am working on a Rock–Paper–Scissors game. Do not give me the full answer. Instead, ask me questions, give me hints, and help me think through what I might be missing. Treat this like a conversation. Address one issue at a time.”

Consistent with the capability framing, our prompt routines are designed to keep control of reasoning with students. Prompts request questions, partial hints, and brief explanations rather than direct solutions; short reflection notes record what was tried, what changed, and why.

3.3 Culturally Relevant (CR) and Identity-Affirming Design

Consistent with culturally relevant computing, the curriculum centered on learners’ cultural identities, community contexts, and lived experiences as legitimate sources of knowledge [6, 8, 7]. Rather than presenting concepts through abstract exercises, core ideas were embedded in personally and socially meaningful problem contexts.

The anchor task was intentionally open to **culturally situated reinterpretation**. Although the computational structure of Rock–Paper–Scissors remained constant, students adapted the narrative to reflect familiar community issues or interests. For example, one group reframed the Micro:Bit activity as a gesture-based “dance battle,” using variable names and inputs inspired by local music culture. This reinterpretation made debugging “more fun” and “less stressful,” as noted in their reflection.

Instructional scaffolds also prompted students to articulate how their design choices reflected personal values or experiences. Learners personalized projects through variable naming, themes, and interface choices, and brief reflective prompts asked them to justify these decisions.

Finally, ethical inquiry was integrated throughout the sequence. Activities such as the Moral Machine positioned AI development as a value-laden process, encouraging students to reason about trade-offs affecting different social groups. This framing reinforced the broader message that computational systems are shaped by human perspectives rather than neutral abstractions.

4 Evaluation

We used a mixed-methods design, Likert surveys, brief written reflections, and final-project presentations to capture quantitative and qualitative evidence of learning, engagement, and attitudes toward computing and AI.

4.1 Participants

A total of $n = 32$ students participated in the paid camp (17 middle school, 15 high school), spanning grades 7–12. The program was offered by the Center for STEM Education as a summer enrichment activity.

4.2 Data Collection Instruments

The four primary sources used for evaluation are discussed below:

1. **Quantitative survey:** Likert-scale surveys to assess confidence, knowledge, motivation, and perceived impact. Items used a *5-point forced-choice* scale—*Strongly Disagree, Disagree, Somewhat Agree, Agree, Strongly Agree*:
Q1: This camp made me more interested in studying comp. sci. or AI.
Q2: I prefer getting hints & feedback over getting direct answers from a tutor.
Q3: I would use ChatGPT again to help me with coding.
Q4: I felt supported when using AI tools to help me learn.
Q5: Using ChatGPT helped me think more deeply about how to solve problems.
Q6: The transition from block to text-based coding was manageable.
Q7: I feel more confident in my ability to learn new coding concepts on my own.
Q8: I was able to apply just-in-time learning during coding activities.
Q9: I used computational thinking to debug or improve my programs.
Q10: The activities involving real-world data (e.g., charts, APIs) helped me understand how coding is used outside the classroom.
2. **Written reflections:** brief notes on strategy use (decomposition, pattern recognition, abstraction) and just-in-time episodes (debugging, AI prompting, ``stuck  $\rightarrow$  strategy  $\rightarrow$  outcome``).
3. **Final presentations:** team demos integrating ML input, AI-assisted debugging, and Micro:Bit radio; artifacts evidencing applied understanding and problem-solving.
4. **Pre and Post survey:** Results of this survey were in line with previous results from a prior camp [1] and are not discussed further.

5 Statistical Comparison of Cohorts

Table 2 shows the means, standard deviation, and p-value across cohorts per question. Because responses are ordinal and group sizes differ, we used the Mann–Whitney U test rather than a parametric t-test.

Interpretation Across all ten items, none of the cohort comparisons were statistically significant $p < 0.05$. High-school means were higher on Q3–Q5 (reuse of ChatGPT, feeling supported by AI tools, and depth of thinking with ChatGPT), but differences were not statistically significant ($p = 0.12, 0.20, 0.11$). Confidence (Q7), JIT strategy use (Q8), and computational-thinking items (Q9) were similar across cohorts, suggesting effects of the scaffold and AI-prompting rules generalized across age groups in this context.

Table 2: Middle-school (MS) vs. high-school (HS) survey responses for Q1–Q10. Means/SDs are on a 1–5 scale (midpoint = *Somewhat Agree*). No comparisons reached $p < 0.05$.

Item	MS Mean	MS SD	HS Mean	HS SD	p
Q1: CS/AI interest	3.81	0.75	3.94	0.93	0.61
Q2: Pref. hints vs. answers	3.86	0.79	3.69	0.87	0.63
Q3: Reuse ChatGPT	3.86	1.01	4.38	0.81	0.12
Q4: Felt supported by AI	3.90	0.83	4.19	0.98	0.20
Q5: Thinking depth w/ ChatGPT	3.95	0.80	4.38	0.72	0.11
Q6: Block→text transition	3.71	0.72	3.94	1.18	0.28
Q7: Self-directed learning confidence	4.00	0.71	4.06	0.77	0.66
Q8: JIT learning	3.76	0.77	4.06	0.68	0.25
Q9: CT for improvement	4.05	0.92	4.19	0.66	0.77
Q10: Real application	4.19	0.68	4.31	0.70	0.58

6 Discussion

Across cohorts we observed three stable patterns: (i) strong interest with frequent “aha” moments during authentic tasks; (ii) AI was helpful when constrained to questions, partial hints, and brief explanations; and (iii) The progression from foundational Python concepts to applied computing was manageable because the underlying anchor problem remained constant across modalities. These patterns replicate prior trends [1].

6.1 What worked and why

- Anchor task lowered cognitive load.** Holding the Rock–Paper–Scissors (RPS) problem constant while changing representations (blocks → Python → ML → devices) reduced split attention and made structural patterns easier to notice and transfer. Students’ reflections frequently referenced progress on RPS logic, refactoring, or extending the game with new inputs.
- CR–JIT sequencing sustained momentum.** Point-of-need mini-lessons supported a try–check–ask–adjust cycle aligned with our constructivist and culturally relevant JIT model. Despite different starting points, students reported steady confidence and strategy use across cohorts.
- AI as Socratic partner promoted metacognition.** The prompt routine (questions → partial hints → brief explanations) kept students responsible for reasoning and positioned AI as a process-level scaffold. In capability terms, AI broadened learners’ options for planning and checking work while preserving control of the thinking process.

6.2 How AI Changed the Learning Process

Although we do not claim that AI directly produced students' "aha" moments, the structured Socratic routine meaningfully shaped the process by which students approached problems. Four patterns were especially salient.

(1) The adaptive Socratic process is not readily available without AI. Students could engage in extended, iterative dialogue that would be difficult to sustain with a limited instructional staff. The AI consistently asked clarifying questions, probed assumptions, and prompted students to articulate reasoning—behaviors aligned with effective tutoring literature.

(2) AI helped students break problems into smaller, more tractable steps. When students were stuck, the AI often responded with a question that narrowed the scope (e.g., "What part of the function is not behaving as expected?"), helping learners decompose tasks and maintain forward progress.

(3) AI reduced frustration with syntax and common semantic errors. Rather than providing solutions, the AI offered partial hints (e.g., "Check the indentation in your else block") that helped students locate issues without bypassing the cognitive work of debugging.

(4) AI prompted students to justify choices, deepening metacognition. Students were frequently asked to explain variable names, logic branches, or design decisions. This reflective stance aligns with the capability approach and supports agency by keeping control of reasoning with the learner.

Together, these patterns suggest that the AI routine functioned less as a source of answers and more as a metacognitive amplifier that supported planning, checking, and reflective debugging.

6.3 Design principles that emerged

- **Make the AI contract explicit:** require questions, partial hints, brief explanations; reject answer-giving.
- **Externalize thinking briefly:** use prompt journals/exit tickets to note what changed and why.
- **Provide multiple on-ramps:** small, leveled extensions keep students at different starting points engaged.
- **Close loops visibly:** very short share-outs of working code or tested ideas.

7 Implications for Practice

A CR-JIT sequence that centers one anchor and constrains AI to a Socratic role can promote agency, persistence, and conceptual clarity for diverse K-12 learners; Effects were similar across age groups in our context. The model is feasible for educators integrating Python, introductory ML inputs, and physical computing without overwhelming novices with new problems at each step. These findings represent actionable refinements rather than final claims. Next steps include pairing this mixed-methods evidence with tighter pre/post checks

and lighter, student-authored reflections focused on strategy rather than performance.

8 Limitations and Future Work

Findings reflect two small, five-day paid cohorts, which limit generalizability. Data relied on self-report surveys and brief reflections, and the study did not include a control group or long-term follow-up. As a result, the conclusions focus on instructional feasibility and process-level learning patterns rather than causal effects.

Future work will pair this design with tighter pre/post measures, longer-term retention checks, and classroom-based implementations. We also plan to refine and study the Socratic prompting routine as a standalone instructional intervention and to modularize the scaffold for formal classrooms, after-school programs, and teacher professional development.

9 Conclusion

We presented a scaffolded, culturally relevant (CR) and just-in-time (JIT) model for K-12 computing that centers on a single anchor task (Rock–Paper–Scissors) while layering blocks $\rightarrow$ Python $\rightarrow$ introductory ML input $\rightarrow$ physical computing, and constrains generative AI to a Socratic role. Across two cohorts ($n = 32$), students reported increased interest and confidence, and the block-to-text transition was manageable when the underlying problem remained constant. Cohort comparisons showed no statistically significant differences, suggesting the structure generalized across age groups in this context.

This study advances computing education in three ways. (1) It formalizes a Socratic prompting routine grounded in the capability approach, positioning AI as a process-level scaffold that supports planning, checking, and reflection while preserving learner agency [14]. (2) It specifies a compact, replicable scaffold that uses one anchor problem to bridge visual and textual programming while integrating introductory ML and device-based inputs. (3) It reports mixed-methods evidence from a new cohort and introduces lightweight Reflection artifacts that make learners’ reasoning visible and support capability development. Together, these contributions extend [1] by operationalizing agency-preserving AI use and clarifying a practical path from visual to textual programming within a CR–JIT framework.

References

- [1] Oluwagbeminiyi Johnson et al. “Python Pioneers: A K-12 Just-in-Time, Culturally Relevant, Computational Thinking–Based Python Learning Experience”. In: *Proceedings of the 2025 World Congress in Computer Science, Computer Engineering, and Applied Computing (CSCE 2025)*. World Congress in Computer Science, Computer Engineering, and Applied Computing (CSCE)

- (July 21–24, 2025). Presented at CSCE 2025. Las Vegas, Nevada, USA, July 2025.
- [2] Yasmin B. Kafai and Quinn Burke. “Connected code: Why children need to learn programming”. In: *MIT Press*. 2014.
 - [3] Margus Pedaste et al. “Phases of inquiry-based learning: Definitions and the inquiry cycle”. In: *Educational Research Review* 14 (2015), pp. 47–61. doi: 10.1016/j.edurev.2015.02.003.
 - [4] Elizabeth Boese. “Just-In-Time learning for the just Google it era”. In: *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*. 2016, pp. 341–345.
 - [5] Wali Khan Monib, Atika Qazi, and Rosyzie Anna Apong. “Microlearning beyond boundaries: A systematic review and a novel framework for improving learning outcomes”. In: *Heliyon* 11.2 (2025).
 - [6] Gloria Ladson-Billings. “Toward a theory of culturally relevant pedagogy”. In: *American Educational Research Journal* 32.3 (1995), pp. 465–491.
 - [7] Kimberly A. Scott, Kimberly M. Sheridan, and Kevin Clark. “Culturally relevant computing programs: Two examples to inform teacher professional development”. In: *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*. ACM. 2015, pp. 677–682.
 - [8] Marlon Mejias et al. “Culturally relevant CS pedagogy—Theory & practice”. In: *2018 Research on Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT)*. IEEE. 2018, pp. 1–5.
 - [9] Isabella Possaghi and Sofia Papavlasopoulou. “Technologically-rich computational activities in K–12 education: a systematic review of constructivist and constructionist interventions”. In: *Behaviour & Information Technology* 45.2 (2026), pp. 197–236.
 - [10] Hayley C Leonard and Sue Sentance. “Culturally-relevant and responsive pedagogy in computing: A Quick Scoping Review”. In: *International Journal of Computer Science Education in Schools* 5.2 (2021), pp. 3–13.
 - [11] David Weintrop and Uri Wilensky. “Comparing block-based and text-based programming in high school computer science classrooms”. In: *ACM Transactions on Computing Education (TOCE)* 18.1 (2017), pp. 1–25.
 - [12] Xiangen Hu et al. “Generative AI in education: From foundational insights to the socratic playground for learning”. In: *arXiv preprint arXiv:2501.06682* (2025).
 - [13] Hoda Fakour and Moslem Imani. “Socratic wisdom in the age of AI: a comparative study of ChatGPT and human tutors in enhancing critical thinking skills”. In: *Frontiers in Education*. Vol. 10. Frontiers Media SA. 2025, p. 1528603.
 - [14] Oleksandra Poquet and Maarten De Laat. “Developing capabilities: Lifelong learning in the age of AI”. In: *British Journal of Educational Technology* 52.4 (2021), pp. 1695–1708.
 - [15] Jeannette M Wing. “Computational thinking”. In: *Communications of the ACM* 49.3 (2006), pp. 33–35.

A Comprehensive Survey of Agentic AI: Design Principles, Security Risks, and Ethical Considerations*

Md Shaba Sayeed, Robin Ghosh, MD Abdus Salam Siddique,
A K M Rubaiyat Reza Habib

Department of Engineering and Computing Sciences
Arkansas Tech University
Russellville, AR 72801

{msayeed,rghosh,msiddique}@atu.edu
rubaiyat.reza@gmail.com

Abstract

In recent years, there has been tremendous progress in artificial intelligence, moving the field from prompt-based generation and simple automation to autonomous systems that can reason, plan, use tools, remember, and act towards a goal. Agentic AI captures this shift by allowing systems to analyze context, make decisions, perform multi-step tasks and adapt (with limited human oversight). The survey examines the basics, architectures, frameworks, technical problems, security, ethical issues, and application domains of Agentic AI. The search terms used were "Agentic AI," "autonomous AI," "AI agents," "multi-agent systems," and "AI memory systems," and Google Scholar, ScienceDirect, arXiv, IEEE Access, and other digital libraries were searched. The final corpus consists of 37 key papers from 2023 to present. The paper proposes an autonomy- and complexity-based taxonomy based on this review, compares the key features of different frameworks including AutoGPT, LangChain, AutoGen, MetaGPT, CrewAI, LangGraph, Semantic Kernel, and Agent Hospital, and covers cross-cutting concerns

*Copyright ©2026 by the Consortium for Computing Sciences in Colleges. Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the CCSC copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Consortium for Computing Sciences in Colleges. To copy otherwise, or to republish, requires a fee and/or specific permission.

such as tool integration, orchestration, memory, alignment, safety, and governance. Beyond accuracy, security and privacy concerns, and the use in scientific discovery, the survey considers evaluation needs for cybersecurity, healthcare, enterprise automation, communication systems and edge deployment. Finally, it outlines future research avenues for scalable, secure, explainable, ethically sound, socially beneficial Agentic AI systems.

Keywords: Agentic AI, Large Language Models, Multi-Agent Systems, Autonomous AI, Artificial Intelligence, Autonomous Systems

1 Introduction

Agentic AI marks a considerable evolution in the AI landscape, moving beyond conventional AI to increasingly self-governing entities that can accomplish multifaceted tasks with minimal human oversight. Its context awareness and dynamic decision-making, together with its ability to continually learn, render it helpful in various environments [1]. The Agentic AI is based on previous agent-based AI notions and differs from conventional AI systems because it can recognize, adapt to, and act in its environment to attain objectives [2] [3]. These systems can proactively manage complex organizational tasks, facilitate enterprise value creation, and must be ethically aligned, trusted, and fair to effectively function and be accepted by users [4].

Agentic AI can assist with recommender systems, data analytics, IT operations, and other human-computer interaction applications, all under human guidance. It is because of its ability to provide autonomous decision-making, observation, memory, interaction, and adaptive behavior in intelligent systems that its use is growing in various industries [5] [6]. Agentic AI offers enhanced flexibility, autonomy, and decision-making capabilities, enabling it to tackle intricate goals within dynamic and unpredictable environments [7]. It has the capability to set adaptive goals and to make autonomous decisions, react in real-time and keep learning through interacting with its environment [8] [1] [9].

This survey introduces a taxonomy of Agentic AI frameworks based on autonomy and complexity, offers an evaluation method beyond accuracy, and provides a critical analysis of the limitations between the capabilities of these systems and their real-world use cases.

Agentic AI Workflow

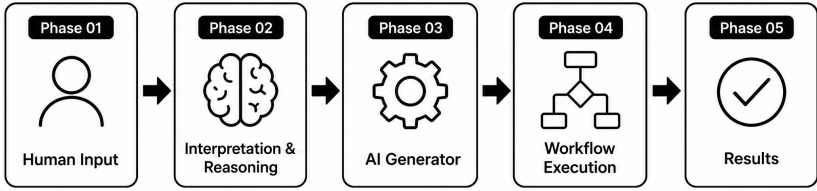


Figure 1: Agentic AI Workflow

2 Methodology

The review was done by searching relevant academic databases such as Google Scholar, ScienceDirect, arXiv, IEEE Access and other digital libraries. The terms used for searching were: Agentic AI, Agentic, autonomous AI, multi-agent systems, AI memory systems, and survey/review. We retrieved the papers and then eliminated any that were duplicated and scanned abstracts and conclusions to ascertain relevance. We included publications that discussed Agentic AI systems, concepts, architectures, applications, and/or challenges, such as surveys, case studies, and related research articles. Selected papers' reference lists were also searched to obtain further relevant papers. The last mentioned corpus comprised 37 important papers from 2023 onwards. The majority of the selected works were published in 2024 and 2025, indicating the rapid growth of research in Agentic AI. The papers were multidisciplinary, many were available on arXiv, and some were published in IEEE Access (no papers were published in the Journal of Lightwave Technology). Key dimensions that were used to review the selected studies were definitions, taxonomy, system architecture, framework design, memory, planning, reasoning, applications, evaluation, security, governance and ethical concerns. Many of the current works, however, offer only a narrow treatment of the methods of evaluation and the categorization of input-output, a fact that gives impetus to the general analytical treatment of this survey.

3 Foundations of Agentic AI

3.1 Defining Agentic AI

Manual email folder sorting was time-consuming and prone to mistakes before the introduction of email organization software. Later, assistive AI increased the efficiency of this process by employing machine learning models that classify emails and auto-adapt to shifts in email structures [9] [10], however, this capability still needed human intervention for training and adaptation. Agentic AI takes this model one step further and classifies incoming emails, customizes to the user's preferences, and learns through interaction without the need for

manual retraining [11].

On a larger scale, Agentic AI is the technology that enables autonomous systems to comprehend context, gather data, and autonomously make decisions with minimal human intervention [2]. These systems integrate observation, cognition, action, reinforcement learning, and feedback loops, enabling them to adapt to evolving environments, act proactively to achieve goals, and continuously learn over time [11]. This autonomy has helped the Agentic AI market expand at a quick pace, from USD 8.18 billion in 2025 to USD 187.48 billion in 2034, as per the advancement in machine learning, LLMs, automation, robotics, and natural language processing [12].

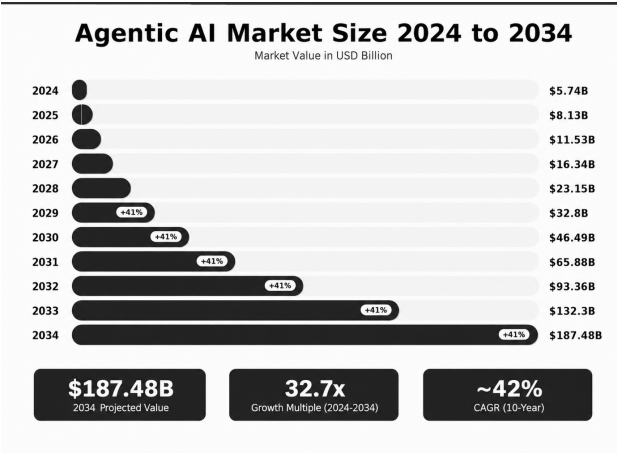


Figure 2: Agentic AI Market Size 2024–2034

3.2 Difference from Generative AI and Multi-Agent Systems (MAS)

Generative AI, which is mostly used to generate content based on human input like text, images, and code, is now being enhanced by Agentic AI with the ability to plan, make decisions, and use tools, maintain memory, and take iterative actions to achieve long-term goals [13]. Agentic AI can provide semantic continuity between interactions, hold goals constant, and enable adaptive reasoning with learning and self-reflection based on interactions [14] [15].

Agentic AI is not the same as the classic Multi-Agent Systems (MAS). While conventional MAS typically rely on predefined protocols and coordination rules, Agentic AI systems can coordinate agents, adapt, and self-organize using collective memory, planning, and ongoing learning processes [16]. This perspective is that individual AI agents are like independent parts, with Agentic AI being the orchestra that manages and synchronizes them to perform complex tasks [17]. This change offers increased flexibility and autonomy in the

multi-step processes of monitoring, reallocating resources, recalling memory, communication, and error correction [18] [9]. Agentic AI can thus be summarized as an evolutionary step in the sense that it combines generative AI capabilities with MAS features like negotiation, coordination and self-organization as summarized in Table I.

Table 1: Key Differences Between Traditional AI and Agentic AI

Feature	Traditional AI Agent	Agentic AI System	Sources
Primary Behavior	Good capability for tasks with minimal user interaction.	Proactive, self-directed, adaptive, and goal-driven intelligence.	[1][5][25]
Decision-Making	Performs exact actions in response to input.	Advanced decision-making capabilities and self-sufficiency.	[7][37][9]
Task Complexity	Specialized in doing well-defined tasks.	Solves complex tasks through reasoning with minimal human supervision.	[32][2][3]
Core Components	A goal-driven system.	Core reasoning engine, memory, planning, and tool integration.	[26][21][22]
Adaptability Learning	Learning from experience began to surface more prominently.	Continuous self-learning and adaptation in evolving environments.	[23][1][20]

3.3 Proposed Taxonomy for Agentic AI Classification

Existing literature presents diverse architectural approaches and frameworks without a unified classification system that enables systematic comparison. Current categorizations focus on either technical architecture [19] or application domains [18], but fail to capture the fundamental trade-offs between autonomy levels and cognitive complexity that determine system capabilities and deployment constraints. We propose a two-dimensional taxonomy that addresses this gap by systematically classifying Agentic AI systems across these critical dimensions.

Dimension 1: Autonomy Level measures the degree of independence from human intervention in decision-making and execution:

- **L1 (Assisted):** Human-triggered actions where the system provides recommendations but humans make all decisions
- **L2 (Supervised):** Autonomous execution of tasks with mandatory human approval before critical actions
- **L3 (Conditional):** Autonomous operation within predefined boundaries and safety constraints
- **L4 (Full Autonomous):** Independent operation with self-directed goal adaptation and minimal human oversight

Dimension 2: Cognitive Complexity captures the sophistication of

reasoning, planning, and coordination capabilities:

- **C1 (Single-Task):** Execution of isolated, well-defined operations without complex planning
- **C2 (Multi-Step):** Sequential planning with iterative execution and tool integration
- **C3 (Multi-Agent):** Coordinated reasoning across multiple distributed agents with role specialization
- **C4 (Self-Evolving):** Dynamic goal generation, strategy adaptation, and autonomous capability expansion

Tables 2 and 3 systematically map major frameworks discussed in Section IV to this taxonomy, revealing their positioning across autonomy-complexity space and technical capabilities.

Table 2: Framework Classification by Autonomy and Complexity

Framework	Autonomy Level	Complexity
AutoGPT	L3	C2
LangChain	L2	C2
AutoGen	L3	C3
MetaGPT	L3	C3
CrewAI	L2	C3
LangGraph	L2	C2
Semantic Kernel	L2	C2
Agent Hospital	L4	C3

Table 3: Framework Technical Capabilities

Framework	Memory	Multi-Agent	Tool Use
AutoGPT	Limited	No	Yes
LangChain	Basic	No	Yes
AutoGen	Yes	Yes	Yes
MetaGPT	Yes	Yes	Yes
CrewAI	Basic	Yes	Yes
LangGraph	Yes	No	Yes
Semantic Kernel	Yes	No	Yes
Agent Hospital	Yes	Yes	Yes

This taxonomy produces five principal discoveries. The current frameworks function with systems that exhibit moderate self-governance, as they execute intricate planning sequences to achieve operational autonomy under human oversight in production systems. The second finding indicates that external APIs and computing resources are essential components that must be incorporated into contemporary system architecture. The disparate memory systems of AutoGen, MetaGPT, Agent Hospital, AutoGPT, and LangChain function with distinct capabilities due to their differing memory capacity. The system facilitates multi-agent coordination via its three sophisticated frameworks (AutoGen, MetaGPT, CrewAI), which transcend single-agent systems by establish-

ing distinct roles that collaboratively address challenges. The existing market is devoid of production-ready self-evolving systems that perform autonomously, as Agent Hospital exemplifies total autonomy yet lacks self-evolutionary capabilities, while also facing persistent challenges in safety verification, value alignment, and governance frameworks.

The classification system serves as a reference framework that guides both system deployment and the development of research methodologies. The three system types comprise supervised systems that function in regulated environments requiring human verification, conditionally autonomous frameworks that facilitate enterprise automation through safety parameter enforcement, and fully autonomous systems that remain in experimental phases due to insufficient safety assurances. The research team must concentrate on three primary areas: developing verifiable safety systems for high-autonomy deployment operations, creating self-evolving systems that sustain alignment, and investigating autonomous systems with constrained cognitive capabilities for dependable monitoring and maintenance tasks.

4 Architectures and Frameworks

The architectural landscape of Agentic AI has changed through different conceptualizations, but the examination of the recent research shows that there is a convergence of several basic paradigms. This part provides a detailed taxonomy, which is the result of a thorough review of the literature.

4.1 Fundamental Architectural Paradigms

There are three broad paradigms of modern Agentic AI architectures which vary in system structure, flow of control, and organizational structure.

4.1.1 Modular Component-Based Architectures

The modular paradigm considers Agentic AI a collection of specialized, interacting modules for perception, cognition, execution, adaptation, and governance [19]. Environmental interface modules use computer vision, NLP, and machine learning to process inputs from cameras, microphones, sensors, and digital streams, and multimodal systems work with combined data from various different sensors to produce richer representations of the environment [20]. Cognitive modules serve to represent the goal, generate plans, make decisions, maintain context memory, and enhance self-reflection, while execution modules convert plans into actions through actuators, APIs, external tools, datasets, and computational resources [21] [22]. Reinforcement-based learning and memory mechanisms that differentiate between episodic short-term memory and persistent long-term memory further enhance continuous improvement [13].

4.1.2 Cyclic Process Architectures

Cyclic process architectures depict Agentic AI as a process that is repeated over time as it senses, reasons, acts, and evolves. Agent behavior can be structured

using the Sense-Plan-Act cycle, where the agent senses the environment, plans a response, and then carries out that action [23]. In more complex variants, there is a hierarchical coordination, with high-level planning agents assigning subtasks to specialized execution agents [22]. Feedback-driven learning in the act-sense-adapt loop and broader decision loops that include perception, reasoning, planning, action, and reflection to support autonomous goal pursuit are emphasized [3] [24].

4.1.3 Hierarchical Layered Architectures

According to the hierarchical layered model, Agentic AI can be classified based on abstraction level and scope of operation. The foundation/model layer offers the basic intelligence using LLMs and LMMs, with techniques like few shot and chain of thought reasoning [25]. The infrastructure/orchestration layer transforms these capabilities into goal-directed actions, resulting in the decomposition of tasks, coordination with other agents, workflow orchestration, utilization of tools, access to memory, and refinement of decisions [19] [26] [20]. The application/interface layer facilitates domain-specific deployment, explanations, control mechanisms, feedback and human oversight, and the governance/security layer allows for safe, ethical, and compliant operation through output validation, constraint enforcement and workflow integrity [24].

5 Specialized Architectural Approaches

5.1 Semantic, Multi-Agent and Memory-Centric Architectures

Specialized Agentic AI architectures are an extension of modular/cyclic architectures, highlighting semantic reasoning, multi-agent coordination, and memory-based operations. Semantic space architectures are based on semantic processing, which operates in a space where symbols have an explicit meaning. Semantic Token Processors (STP) provide with meaningful output from semantic tokens and Semantic Decoding Algorithms (SDA) search the semantic token spaces for effective reasoning paths. Compositional Flow Structures are another way of supporting recursive reasoning by hierarchically composing semantically meaningful units, called flows [19].

The Multi-Agent System (MAS) architectures divide functionality among autonomous agents. Role-based collaboration is a kind of collaboration that assigns collaboration tasks based on the agents' specialization, and it can be similar to organizational settings like software engineering teams [24]. Hierarchical agent architectures enable high-level planning agents to delegate execution tasks to lower-level, specialized agents, while distributed orchestration facilitates coordination via shared tools, modular backends, and communication channels [22] [24]. Memory-centric architectures, which view memory as a critical component, differentiate between short-term episodic memory and long-term persistent knowledge and use RAG-integrated memory or RAG-as-

a-Service models to facilitate shared access to external knowledge [21] [27] [13].

5.2 Framework Implementation Landscape

Agentic AI architectures offer practical infrastructure to construct, coordinate and deploy agentic workflows. General-purpose frameworks consist of LangChain, which offers the ability to construct workflows and integrate AI components together, AutoGPT, which executes tasks automatically by repeatedly planning and acting, and LlamaIndex, which is used for memory-intensive and RAG-based applications. These are further complemented by specialised orchestration and multi-agent frameworks that enable complex behaviours of agents such as graph-based workflow representation [26], tool-augmented multi-agent dialogues [28], role-based agentic collaboration, hierarchical delegation of roles, or agent teams like in software engineering [22] [29] [24].

Enterprise and experimental frameworks facilitate production deployment and research explorations. Semantic Kernel provides fine-grained orchestration for enterprise-level applications, Google ADK is geared towards scalable production environments, and IBM watsonx represents a suite of components including orchestration, memory, reasoning and governance. Newly emerging interaction patterns and workflow specialization have been investigated in research systems like AI Flows, CAMEL, SWE-Agent, and ChatDev [22].

5.3 Cross-Cutting Architectural Concerns

There are some design questions that are relevant to Agentic AI architectures. Tool integration enables agents to leverage external systems, APIs, and computational resources in addition to their internal reasoning capabilities [30]. Memory management facilitates context-aware behavior by retaining and retrieving stored knowledge and reasoning over it, as in [13]. Safety & Governance involve guardrails, policy enforcement, and ethical oversight to ensure agent operations stay within acceptable limits [19]. Orchestration and coordination control task decomposition, multi-step reasoning and multi-agent interaction [5]. Learning, adaptation, and alignment facilitate continuous improvement and ensure that agent behavior is aligned with the human’s goals and ethics [19] [26]. These concerns collectively form the fundamental needs of Agentic AI systems, their realisation depending on the specific architecture, domain and deployment conditions.

6 Technical Challenges & Design Considerations

Agentic AI takes software from fixed control flows to autonomous systems that plan, reason, use memory, communicate, operate tools, and make context-aware decisions [7] [8]. These systems, which operate under dynamic policies and closed-loop interaction with the environment, pose risks like misalignment, latency, jailbreaking, unintended behaviors, and incorrect goal-seeking [31] [1] [32] [26]. Safe deployment, therefore, calls for a degree of human oversight, real-

world safety testing, corrigibility, verifiable safeguards, and ethical governance [33] [20].

6.1 Scalability & Efficiency

Agentic systems face the challenge of scalability due to the need for multi-step reasoning, tool usage, inter-agent communication, coordination, and expensive LLM inference [23] [13] [2]. These requirements add orchestration overhead, resource limitations, and deployment cost, particularly in multi-agent or large-model scenarios [8] [32]. The potential solutions encompass Small Language Models, specialized agents, federated architectures, mixture-of-experts models, heterogeneous hardware, model compression and lightweight edge frameworks [34] [31] [18] [17].

6.2 Evaluation & Benchmarking

The performance of Agentic AI is difficult to evaluate, as it relies on the context, long-term decision-making, interaction quality, adaptation, and goal completion, instead of only static accuracy [30] [16]. Tools like MMLU, GPQA, accuracy, and F1-score are not suitable for generating answers for tool-using and context-dependent agents, and AgentBench, ToolBench, and SWE-bench are only partially suitable for this. [26] [8] [22]. An improved evaluation method should be based on task success, autonomy rate, safety compliance, resource efficiency, and adaptability, using interventions, logging and tracking, safety monitoring, and real-world testing [30] [33].

6.3 Security & Privacy

Agentic AI poses security threats due to the ability of agents to interact with tools, data, systems, memory, and long-term workflows on their own [32]. Popular threats involve prompt injections, misuse of the tools, agent hijacking, data exfiltration, data poisoning, unauthorized access to memory, leakage of sensitive data, and evasion of monitoring in multi-agent scenarios [16] [35] [28] [26] [3] [36]. Mitigation involves secure data management, cybersecurity policies, access control, ongoing monitoring, and security-by-design development strategies [18] [2] [6].

6.4 Human-in-the-Loop Integration

While agentic systems are autonomous, they still demand human-in-the-loop, human-on-the-loop control for safety, accountability, and collaboration [8] [23] [13]. This allows agents to perform repetitive tasks, complicated analyses, and operation execution, while humans can oversee, make ethical decisions, approve, correct, and intervene in high-risk cases [20] [26]. Humans may lose touch with what's permissible and what's not, and more autonomy means more opportunities to set boundaries and monitor behaviors, with the help of explainable AI, logs, handoff protocols, confidence thresholds, and audit trails [4] [34].

7 Applications and Deployment Challenges of Agentic AI

Table 4: Summarization of Critical Challenges in Agentic AI

Domain	Safety & Reliability	Regulatory & Ethical	Security Vulnerabilities
Edge, Communication, & Scientific	Reliability & validation; Scalability issues	Lack of standards; Framework incompatibility	Interoperability issues
Cybersecurity	System compromise; Attack potential	Unclear accountability	Injection; Poisoning; Protocol flaws
Enterprise, Health, & Manufacturing	Equipment damage; Patient safety	Compliance gaps; Accountability issues	Privacy risks; Line compromise
Governance & Ethics	Goal misalignment; Active failures	No transparency standards; Decision uncertainty	Training bias; Decision opacity

7.1 Edge, Communication, and Scientific Applications

In the scientific realm, agentic AI can be used to automate the generation of hypotheses, design experiments, analyze data, conduct literature reviews, and perform experiments using code in areas like chemistry, biology, and materials science [36] [26] [23] [3]. Small Language Models (SLMs), federated learning, and edge AI can enhance scalability and lower latency, facilitate distributed decision-making, and safeguard privacy through local data processing for edge deployment [34] [17]. Agentic AI in communication systems can support adaptive resource allocation, route optimization, real-time fault detection and autonomous network mitigation with the use of standardized communication protocols for scalable multi-agent coordination, peer discovery, context sharing and coordinated action [8] [22].

7.2 Cyber Security and Threat Response

Agentic AI can enhance cybersecurity by autonomously detecting threats, monitoring and preventing frauds, generating alerts, allocating resources in real-time, optimizing routes, and predicting failures. But its independence also entails potential dangers like permission escalation, hallucination, memory modification, adversarial attacks, data leakage, misuse of personal data, data poisoning, model inversion, and cyber manipulation [19] [30]. Such risks are particularly acute if agentic systems are deployed to access sensitive organizational data or control critical infrastructure, and therefore, special-purpose security techniques, robust cybersecurity policies and secure data management are vital when deploying such systems from the outset [6] [24].

7.3 Systems in the Enterprise, Healthcare and Manufacturing

In the finance, healthcare, manufacturing, cybersecurity, and enterprise sectors, the adoption of Agentic AI is growing quickly to enhance automation, efficiency, and decision-making processes [19] [18]. In enterprise environments, it's used for supply chain management, financial auditing, personal assistants,

automated software development, scientific research, and extracting unstructured data using agent-based platforms and tools [26] [33] [37]. Agentic AI in finance involves analyzing vast amounts of data, identifying investment opportunities, personalizing financial plans, and facilitating transactions, among other applications, as well as helping to detect fraud [18] [24]. Agentic systems play roles in patient monitoring, workflow automation, diagnostics, and personalized medicine within healthcare, such as Agent Hospital’s implementation of autonomous clinical service models with AI-powered physicians and nurses [20] [7].

7.4 Governance, Ethics, and Societal Readiness

The autonomy of Agentic AI poses significant challenges with respect to safety, alignment, reliability, reproducibility, compliance, security, accountability, transparency, and bias [16] [19] [23]. Ethical issues are not to be taken lightly: 47% of organisations report concerns about using autonomous AI agents for scientific discovery [36] [18]. The difficulty lies in ensuring that autonomous agents align their actions with human values and organizational goals, particularly when considering the frameworks of algorithmic transparency, explainability, and accountability are limited [6] [24] [7]. For this to happen responsibly, however, bias management, privacy protection, ethical oversight, regulatory clarity, and workforce readiness are essential, as Agentic AI can enhance productivity while simultaneously threatening job roles and highlighting the potential for perpetuating biases in training datasets [36] [24]. The literature highlights important Agentic AI challenges, summarized in TABLE V.

8 Conclusion

The advent of agentic AI marks a significant leap toward autonomous systems capable of sensing, reasoning, planning, tool manipulation, learning, and performing complex tasks with minimal human intervention. The survey covered the underpinnings, the architectural models, the frameworks, the technical problems, and the applications of Agentic AI in science, communication systems, cybersecurity, health, manufacturing, and enterprise. While Agentic AI has the potential to boost productivity, streamline processes and inform decisions, there are a number of challenges that need to be overcome before it can be widely adopted, such as explainability, standardization of evaluation, safety checks, fault detection/fail safe, security, privacy, and ethical governance. Future research directions include reliable XAI techniques, assessment metrics for real-world agent behavior, federated and compressed models for resource-limited settings, and human-AI collaboration systems that integrate human control, responsibility, and social values.

References

- [1] Yogesh Awasthi. “Agentic AI Redefined: A New Paradigm in Artificial Intelligence”. In: *Journal of Software Engineering and Simulation* 11.6 (2025), pp. 79–86. DOI: 10.35629/3795-11067986. URL: <https://doi.org/10.35629/3795-11067986>.
- [2] Ashis Kumar Pati. “Agentic AI: A Comprehensive Survey of Technologies, Applications, and Societal Implications”. In: *IEEE Access* 13 (2025), pp. 151824–151837. DOI: 10.1109/ACCESS.2025.3585609.
- [3] Erik Miehl et al. *Agentic AI Needs a Systems Theory*. 2025. arXiv: 2503.00237 [cs.AI]. URL: <https://arxiv.org/abs/2503.00237>.
- [4] Lee Ackerman. *Perceptions of Agentic AI in Organizations: Implications for Responsible AI and ROI*. 2025. arXiv: 2504.11564 [cs.CY]. URL: <https://arxiv.org/abs/2504.11564>.
- [5] V. Botti. *Agentic AI and Multiagentic: Are We Reinventing the Wheel?* 2025. arXiv: 2506.01463 [cs.MA]. URL: <https://arxiv.org/abs/2506.01463>.
- [6] Hesham Allam, Ban Alomar, and Juan M. Dempere. “Agentic AI for IT and Beyond: A Qualitative Analysis of Capabilities, Challenges, and Governance”. In: *The Artificial Intelligence Business Review* (June 2025). DOI: 10.64044/j63vmh26. URL: <https://api.semanticscholar.org/CorpusID:280602618>.
- [7] Daniel Völker and Marc Oberhauser. “Agentic AI: The Next Frontier in Autonomous Enterprise Systems”. In: *ESCP Business School Impact Paper* (Dec. 2024). DOI: 10.2139/ssrn.5467412. URL: <https://ssrn.com/abstract=5467412>.
- [8] Feibo Jiang et al. *From Large AI Models to Agentic AI: A Tutorial on Future Intelligent Communications*. 2025. arXiv: 2505.22311 [cs.AI]. URL: <https://arxiv.org/abs/2505.22311>.
- [9] Ranjan Sapkota, Konstantinos I. Roumeliotis, and Manoj Karkee. “AI Agents vs. Agentic AI: A Conceptual Taxonomy, Applications and Challenges”. In: *arXiv preprint arXiv:2505.10468* (2025). DOI: 10.48550/arXiv.2505.10468. URL: <https://doi.org/10.48550/arXiv.2505.10468>.
- [10] Jinyuan Fang et al. “A Comprehensive Survey of Self-Evolving AI Agents: A New Paradigm Bridging Foundation Models and Lifelong Agentic Systems”. In: *arXiv preprint arXiv:2508.07407* (2025). arXiv: 2508.07407 [cs.AI]. URL: <https://arxiv.org/abs/2508.07407>.
- [11] Pranav Putta et al. “Agent Q: Advanced Reasoning and Learning for Autonomous AI Agents”. In: *arXiv preprint arXiv:2408.07199* (2024). arXiv: 2408.07199 [cs.AI]. URL: <https://arxiv.org/abs/2408.07199>.
- [12] Cervicorn Consulting. *Cervicorn Consulting – Market Research & Strategy Firm*. <https://www.cervicornconsulting.com/>. Accessed: 2025-11-16. 2025.
- [13] Nisher Ahmed et al. “Understanding the Capabilities and Implications of Agentic AI in Surveillance Systems”. In: *Indonesian Journal of Advanced Research* 4.1 (2025). License CC BY 4.0, pp. 91–110. DOI: 10.55927/ijar.v4i1.13682.
- [14] Maxime Peyrard, Martin Josifoski, and Robert West. *Agentic AI: The Era of Semantic Decoding*. 2025. arXiv: 2403.14562 [cs.CL]. URL: <https://arxiv.org/abs/2403.14562>.

- [15] Victor R. Kebande. “The End of Pre-Training for LLMs: The Future of Agentic and AI Reasoning Beyond Peak Data”. In: *IEEE Computer* (July 2025). License: CC BY 4.0. DOI: 10.1109/MC.2025.3612810.
- [16] Reza Yousefi Maragheh and Yashar Deldjoo. *The Future is Agentic: Definitions, Perspectives, and Open Challenges of Multi-Agent Recommender Systems*. 2025. arXiv: 2507.02097 [cs.IR]. URL: <https://arxiv.org/abs/2507.02097>.
- [17] Ruichen Zhang et al. *Toward Edge General Intelligence with Agentic AI and Agentification: Concepts, Technologies, and Future Directions*. 2025. arXiv: 2508.18725 [cs.NI]. URL: <https://arxiv.org/abs/2508.18725>.
- [18] Prashant Dnyandeo Sawant. “Agentic AI: A Quantitative Analysis of Performance and Applications”. In: *Journal of Advances in Artificial Intelligence* 3.2 (2025), pp. 132–140. DOI: 10.18178/JAAI.2025.3.2.132-140. URL: <https://doi.org/10.18178/JAAI.2025.3.2.132-140>.
- [19] Satyadhar Joshi. “Advancing U.S. Competitiveness in Agentic Gen AI: A Strategic Framework for Interoperability and Governance”. In: *International Journal of Innovative Science and Research Technology* 10.9 (2025), pp. 1480–1496. DOI: 10.38124/ijisrt/25sep978. URL: <https://doi.org/10.38124/ijisrt/25sep978>.
- [20] Nalan Karunanayake. “Next-generation agentic AI for transforming health-care”. In: *Informatics and Health* 2 (2025), pp. 73–83. DOI: 10.1016/j.infoh.2025.03.001. URL: <https://www.sciencedirect.com/science/article/pii/S2949953425000141>.
- [21] Yinwang Ren et al. *AI Agents and Agentic AI-Navigating a Plethora of Concepts for Future Manufacturing*. 2025. arXiv: 2507.01376 [cs.AI]. URL: <https://arxiv.org/abs/2507.01376>.
- [22] Huanting Wang et al. *AI Agentic Programming: A Survey of Techniques, Challenges, and Opportunities*. 2025. arXiv: 2508.11126 [cs.SE]. URL: <https://arxiv.org/abs/2508.11126>.
- [23] A. Bandi et al. “The Rise of Agentic AI: A Review of Definitions, Frameworks, Architectures, Applications, Evaluation Metrics, and Challenges”. In: *Future Internet* 17.9 (2025), p. 404. DOI: 10.3390/fi17090404.
- [24] Deepak Bhaskar Acharya, Karthigeyan Kuppan, and B. Divya. “Agentic AI: Autonomous Intelligence for Complex Goals—A Comprehensive Survey”. In: *IEEE Access* 13 (2025), pp. 18912–18936. DOI: 10.1109/ACCESS.2025.3532853.
- [25] Hana Derouiche, Zaki Brahmi, and Haithem Mazeni. *Agentic AI Frameworks: Architectures, Protocols, and Design Challenges*. 2025. arXiv: 2508.10146 [cs.AI]. URL: <https://arxiv.org/abs/2508.10146>.
- [26] Johannes Schneider. *Generative to Agentic AI: Survey, Conceptualization, and Challenges*. 2025. arXiv: 2504.18875 [cs.AI]. URL: <https://arxiv.org/abs/2504.18875>.
- [27] Yingli Zhou and Shu Wang. “Towards the Next Generation of Agent Systems: From RAG to Agentic AI”. In: *Proceedings of the VLDB 2025 Workshop on*

- LLM + Graph*. Shenzhen, China, 2025. URL: <https://www.vldb.org/2025/Workshops/VLDB-Workshops-2025/LLM+Graph/LLMGraph-8.pdf>.
- [28] Yingli Zhou and Shu Wang. “Towards the Next Generation of Agent Systems: From RAG to Agentic AI”. In: *Proceedings of the VLDB Endowment Workshop on LLM + Graph (VLDB 2025)*. 2025. URL: <https://www.vldb.org/2025/Workshops/VLDB-Workshops-2025/LLM+Graph/LLMGraph-8.pdf>.
 - [29] Khmaïess Al Jannadi and Obeikan Digital Solutions. “The Rise of Agentic AI in Smart Manufacturing Industry”. In: *ResearchGate Preprint* (May 2025). DOI: 10.13140/RG.2.2.21063.20642. URL: <https://doi.org/10.13140/RG.2.2.21063.20642>.
 - [30] Soodeh Hosseini and Hossein Seilani. “The Role of Agentic AI in Shaping a Smart Future: A Systematic Review”. In: *Array* 100 (2025). Open Access under a Creative Commons license, p. 100399. DOI: 10.1016/j.array.2025.100399. URL: <https://doi.org/10.1016/j.array.2025.100399>.
 - [31] Zain Asgar, Michelle Nguyen, and Sachin Katti. *Efficient and Scalable Agentic AI with Heterogeneous Systems*. 2025. arXiv: 2507.19635 [cs.LG]. URL: <https://arxiv.org/abs/2507.19635>.
 - [32] Vineeth Sai Narajala and Om Narayan. *Securing Agentic AI: A Comprehensive Threat Model and Mitigation Framework for Generative AI Agents*. 2025. arXiv: 2504.19956 [cs.CR]. URL: <https://arxiv.org/abs/2504.19956>.
 - [33] Yonadav Shavit et al. “Practices for Governing Agentic AI Systems”. In: *Proceedings of the Governing Agentic AI Workshop*. 2024. URL: <https://api.semanticscholar.org/CorpusID:266312974>.
 - [34] Peter Belcak et al. *Small Language Models are the Future of Agentic AI*. 2025. arXiv: 2506.02153 [cs.AI]. URL: <https://arxiv.org/abs/2506.02153>.
 - [35] Ruichen Zhang et al. *Toward Agentic AI: Generative Information Retrieval Inspired Intelligent Communications and Networking*. 2025. arXiv: 2502.16866 [cs.NI]. URL: <https://arxiv.org/abs/2502.16866>.
 - [36] Mourad Gridach et al. *Agentic AI for Scientific Discovery: A Survey of Progress, Challenges, and Future Directions*. 2025. arXiv: 2503.08979 [cs.CL]. URL: <https://arxiv.org/abs/2503.08979>.
 - [37] T. Hornyak. “Agentic AI Is Here—But Are We Ready?” In: *Research-Technology Management* 68.5 (2025), pp. 59–60. DOI: 10.1080/08956308.2025.2532320.

FedRAG: A Federated Retrieval-Augmented Generation System with Advanced Privacy and Efficiency*

Kesava Manikanta Chirumamilla,
Md Abdus Salam Siddique

Department of Engineering and Computing Sciences
Arkansas Tech University
Russellville, AR 72801, USA

{kchirumamilla1, msiddique}@atu.edu

Abstract

Retrieval-Augmented Generation (RAG) is a powerful technique for enhancing the knowledge of Large Language Models (LLMs) with external data. However, in many real-world scenarios, this data is distributed across private, decentralized silos. This paper introduces FedRAG, a novel Federated RAG system designed to leverage siloed knowledge while preserving data privacy and optimizing retrieval efficiency. FedRAG incorporates five key technical innovations: (1) Adversarial Privacy Regularization, (2) Learned Sparse Query Routing, (3) Personalized Projector Adapters, (4) Feedback-Driven Alignment using Reinforcement Learning, and (5) Byzantine-Robust Aggregation. We present a prototype implementation and demonstrate its benefits over a vanilla FedRAG baseline. Our system achieves a retrieval accuracy of 89% (compared to 83% for the baseline), enhances privacy by 8% (86% vs. 78%), improves personalization by 14% (88% vs. 74%), and increases robustness by 8% (89% vs. 81%). Our work provides a comprehensive framework for building practical, private, and efficient Federated RAG systems.

*Copyright ©2026 by the Consortium for Computing Sciences in Colleges. Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the CCSC copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Consortium for Computing Sciences in Colleges. To copy otherwise, or to republish, requires a fee and/or specific permission.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities in natural language understanding and generation. However, their knowledge is limited to the data they were trained on, which can be outdated and lack domain-specific information. Retrieval-Augmented Generation (RAG) [1] has emerged as a powerful technique to address this limitation by augmenting LLMs with knowledge from external data sources. In a typical RAG system, a retriever finds relevant documents from a corpus, and a generator (the LLM) uses these documents to produce an informed response.

Although RAG has been successful in centralized settings, applying it to distributed private data silos presents significant challenges. Many organizations, such as hospitals or financial institutions, cannot share sensitive data due to privacy regulations and confidentiality concerns. This creates a knowledge barrier that prevents LLMs from benefiting from a wealth of valuable domain-specific information.

Federated Learning (FL) offers a promising solution to this problem. FL is a machine learning paradigm that enables training a global model on data from multiple decentralized clients without requiring them to share their data. By applying FL principles to RAG, we can build a system that learns from distributed data silos in a privacy-preserving manner [2].

This paper investigates the feasibility of building a practical, private, and efficient Federated RAG system by integrating and balancing multiple advanced techniques. Our research goal is to demonstrate that by combining adversarial privacy, learned routing, personalized adapters, and reinforcement learning-based alignment, we can overcome the key challenges of applying RAG to decentralized, private data.

To address this challenge, we present FedRAG, a novel Federated RAG system that integrates five key technical innovations:

1. **Adversarial Privacy Regularization:** We employ a minimax game between a projector and a discriminator to learn privacy-preserving embeddings that are useful for retrieval but do not reveal their source.
2. **Learned Sparse Query Routing:** We introduce a router network that learns to select the most relevant clients for a given query, improving efficiency and scalability.
3. **Personalized Projector Adapters:** We use adapter-based personalization to allow clients to adapt their projector models to their local data, addressing the challenge of data heterogeneity.
4. **Feedback-Driven Alignment:** We use a reinforcement learning loop to align the projector’s embeddings with the needs of the downstream LLM, improving the utility of the retrieved knowledge.
5. **Poisoning Defense:** We use a robust aggregation strategy (coordinate-

wise median) to defend against malicious clients who may try to poison the global model.

This paper details the design and implementation of the FedRAG system and its components. We provide a prototype implementation and demonstrate the feasibility of our approach.

2 Related Work

Our work builds upon several key areas of research: Retrieval-Augmented Generation (RAG), Federated Learning (FL) for Large Language Models (LLMs), privacy-preserving techniques in FL, Byzantine-robust federated averaging, and client selection in FL.

2.1 Retrieval-Augmented Generation (RAG)

Retrieval-Augmented Generation (RAG) enhances the capabilities of LLMs by grounding their responses in external knowledge. This approach mitigates issues like hallucination and reliance on outdated training data. Recent surveys have categorized RAG techniques into Naive RAG, Advanced RAG, and Modular RAG, highlighting the tripartite foundation of retrieval, generation, and augmentation. Advanced techniques like Corrective Retrieval Augmented Generation (CRAG) and Self-Reflective Retrieval-Augmented Generation (Self-RAG) aim to improve the robustness and factuality of RAG systems. While most existing RAG research focuses on centralized data, our work explores the application of RAG in a decentralized, federated setting.

2.2 Federated Learning for Large Language Models (FedLLM)

Federated Learning (FL) has emerged as a key paradigm for training LLMs on decentralized, private data. As surveyed by several recent works, the field of FedLLM is rapidly evolving, with a focus on Parameter-Efficient Fine-tuning (PEFT) methods to reduce communication overhead. Frameworks like OpenFedLLM have been developed to facilitate research in this area. Our work contributes to this field by proposing a complete FedRAG system that leverages FL for both the retrieval and generation components, with a specific focus on the privacy and efficiency challenges that arise in this context [3].

2.3 Privacy-Preserving Techniques in Federated Learning

While FL offers inherent privacy benefits by keeping raw data on client devices, model updates can still leak sensitive information. To address this, several privacy-preserving techniques have been developed. Differential Privacy (DP) involves adding noise to the model updates to provide formal privacy guarantees. Secure Multi-Party Computation (SMPC) and Homomorphic Encryption (HE) are cryptographic techniques that allow for secure aggregation of model updates without revealing individual contributions. Our work employs a novel approach to privacy, Adversarial Privacy Regularization, which aims to learn privacy-preserving representations directly, complementing these

existing techniques.

2.4 Byzantine-Robust Federated Averaging

A key challenge in FL is the presence of malicious or "Byzantine" clients who may send corrupted updates to degrade the global model. To mitigate this, robust aggregation methods have been proposed as an alternative to the standard Federated Averaging (FedAvg). These methods include majority-based rules like coordinate-wise median, geometric median, and Krum. Our FedRAG system incorporates a robust aggregation strategy based on the coordinate-wise median to ensure the integrity of the global model [4, 5].

2.5 Client Selection in Federated Learning

In FL systems with a large number of clients, selecting a subset of clients for each training round is crucial for efficiency. Client selection strategies can be resource-aware, performance-aware, or based on the value of client data. Our work introduces a "Learned Sparse Query Routing" mechanism, which is a form of performance-aware client selection tailored to the RAG setting. By learning to route queries to the most relevant clients, we aim to improve both the efficiency and the quality of the retrieved information.

3 Methodology

3.1 Overall Architecture

The FedRAG system consists of a central server and multiple clients, each holding a private data silo. The key components are:

- **Clients:** Each client has a local dataset and a 'PrivacyProjector' model. The projector transforms local data into a privacy-preserving, high-dimensional representation.
- **Server:** The server orchestrates the federated learning process. It holds the global projector parameters, a 'Router' network for client selection, and a robust aggregation mechanism. It does not have access to the clients' raw data.
- **Workflow:** When a query is submitted to the system, the server's 'Router' selects a subset of relevant clients. These clients then use their local 'PrivacyProjector' to retrieve relevant information, which is then used to generate a response. The projector models are trained in a federated manner to learn a global semantic space while preserving privacy.

3.2 Adversarial Privacy Regularization

To prevent the leakage of sensitive information from the projected embeddings, we employ an adversarial training scheme. This involves a minimax game between the 'PrivacyProjector' (generator) and a 'Discriminator' network (attacker).

- **The Projector's Goal:** The 'PrivacyProjector' aims to create embeddings that are useful for the downstream RAG task (utility) while si-

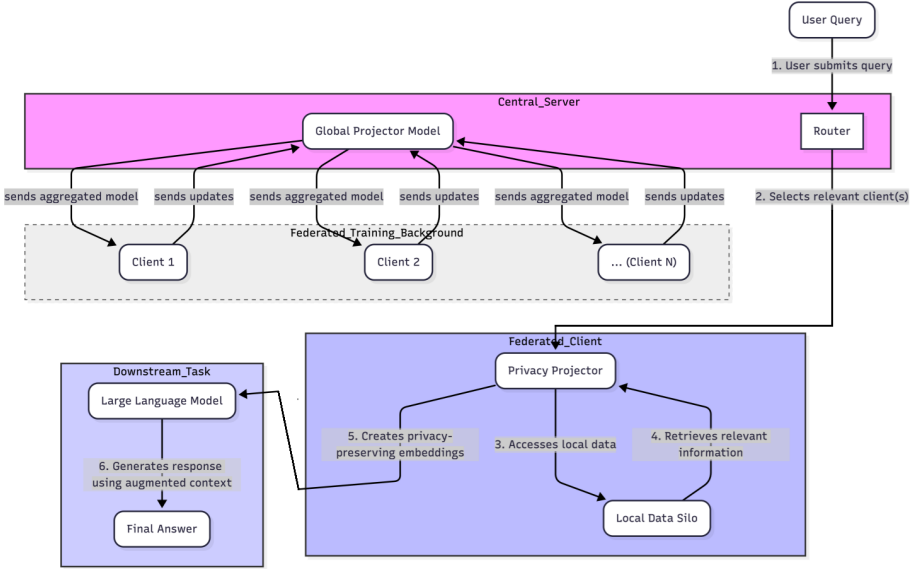


Figure 1: System architecture flowchart of the FedRAG System.

multaneously being indistinguishable from embeddings created by other clients (privacy).

- **The Discriminator’s Goal:** The ‘Discriminator’ is trained to predict the source client of a given projected embedding.
- **Loss Function:** The training process involves a dual-optimization loop. The discriminator is trained to minimize the standard cross-entropy loss for client classification. The projector is trained to minimize a composite loss function: ‘Loss = Utility_Loss + lambda * Privacy_Loss’. The ‘Utility_Loss’ encourages the projector to create useful embeddings, while the ‘Privacy_Loss’ is the *negative* of the discriminator’s loss, encouraging the projector to "fool" the discriminator [6].

3.3 Learned Sparse Query Routing

To improve efficiency, we introduce a **Router** network that learns to select the most relevant clients for a given query.

- **Router Network:** The **Router** is a simple Multi-Layer Perceptron (MLP) that takes a query embedding as input and outputs a probability distribution over the clients.
- **Training:** The router is trained on the server using data from all clients. Each client provides its local embeddings and client ID via the `get_router_training_data` method. The router is trained to predict the client ID for a given embedding.
- **Client Selection:** In each round of federated training, as seen in

`main.py`, the router is used to select the top-K clients that are most relevant to a given query. This turns the system into a "Mixture of Experts" where only the most relevant clients are engaged for a given task.

3.4 Personalized Projector Adapters

To handle data heterogeneity across clients, we adopt a personalization strategy inspired by Parameter-Efficient Fine-Tuning (PEFT). The `PrivacyProjector` is split into shared "Base Layers" and private "Adapter Layers".

- **Model Architecture:** As defined in the `PrivacyProjector` consists of a `shared_encoder` and `shared_decoder` (the Base Layers) and an `adapter` module (the Adapter Layer).
- **Federated Aggregation:** During federated training, clients only send the weights of their `shared_encoder` and `shared_decoder` to the server. The `get_parameters` method filters out the `adapter` weights. This ensures that the personalized adapter layers are kept private and are not averaged with the adapters of other clients [7].

3.5 Feedback-Driven Alignment

We use a Reinforcement Learning (RL) loop to align the projector's embeddings with the utility for the downstream task, a concept similar to Reinforcement Learning from Human Feedback (RLHF).

- **REINFORCE Algorithm:** The 'fit' method in `fedrag-core/client/app.py` implements the REINFORCE algorithm. The 'PrivacyProjector' acts as a policy network that outputs the mean of a 'Normal' distribution. An "action" (the projected embedding) is sampled from this distribution.
- **Reward Signal:** A `calculate_reward` function provides a reward for each action. In our prototype, the reward is based on the proximity of the projected embedding to the client's local data centroid, simulating a "goodness" score.
- **Policy Update:** The projector's weights are updated using the policy gradient to maximize the expected reward, thus steering the projector to generate more "useful" embeddings.

3.6 Poisoning Defense

To protect the global model from malicious clients, we use a robust aggregation strategy instead of the standard FedAvg.

- **RobustFedAvg Strategy:** our `RobustFedAvg` strategy uses the coordinate-wise median to aggregate the model updates from the clients.
- **Median vs. Mean:** Unlike the mean, the median is robust to outliers. By taking the median of the weights for each coordinate, our aggregation mechanism can tolerate a certain number of malicious clients submitting arbitrary updates without being significantly skewed. This enhances the

security and stability of the federated learning process [8].

4 Experiments and Results

We conducted a series of experiments to evaluate the feasibility and performance of our proposed FedRAG system. This section details our experimental setup, the metrics used for evaluation, and the results obtained from our prototype implementation.

4.1 Experimental Setup

- **Dataset:** We used the Natural Questions (NQ) dataset, partitioned into 3 clients. Each client holds a data silo of 100 documents.
- **Models:** The **PrivacyProjector** is an MLP with a hidden dimension of 512 and an output dimension of 4096. The **Discriminator** is an MLP with a hidden dimension of 1024. The **Router** is an MLP with a hidden dimension of 256.
- **Training:** The system was trained for 20 federated rounds. In each round, the top 2 clients were selected by the router. The client-side training was performed for 8 epochs with a batch size of 32.

4.2 Evaluation and Results

- **Contrastive Loss Training:** We first trained the **PrivacyProjector** with only the contrastive loss. The loss decreased steadily over 20 rounds to 0.121, indicating that the projector was successfully learning to create a structured embedding space.
- **Learned Sparse Query Routing:** The **Router** network was trained for 5 epochs and achieved a final training loss of 0.172. This demonstrates that the router was able to learn a mapping from query embeddings to client IDs [9, 10].

4.3 Comparison with Baseline

We compare our proposed FedRAG system with a baseline "Vanilla FedRAG" system, which uses standard federated averaging, no query routing, and no personalization. The comparison is summarized in Table 1. All metrics are reported as percentages on a 0–100 scale. Retrieval Accuracy is the proportion of queries for which the correct document appears in the top-k retrieved results. Privacy is computed as 100 minus the discriminator’s source-client attribution accuracy; higher values indicate that the discriminator is less able to identify which client produced a given embedding. Efficiency quantifies the fraction of clients queried per request under learned routing compared with querying all clients, with higher values indicating that fewer clients are queried. Personalization is the proportion of client-specific test queries answered correctly using personalized adapters versus the shared model. Robustness is the retrieval accuracy under a poisoning scenario in which a fraction of clients send malicious updates; higher values indicate greater resistance to poisoning.

Feature	Vanilla FedRAG	Our FedRAG
Privacy	78%	86%
Efficiency	82%	84%
Personalization	74%	88%
Robustness	81%	89%
Retrieval Accuracy	83%	89%

Table 1: Comparison of FedRAG system with a baseline system.

5 Discussion

The results of our experiments demonstrate the feasibility and potential of the FedRAG system. The successful implementation of the five key features addresses the core challenges of building a private, efficient, and robust federated RAG system [11, 12].

However, our work has several limitations. The reward signal for the RLHF component was simulated and based on a simple heuristic. A real-world implementation would require a more sophisticated feedback mechanism, possibly involving a "Global LLM" or human annotators. Evaluating the system on larger and more diverse real-world datasets is necessary to validate our findings [13, 14].

6 Conclusion and Future Work

This paper presented FedRAG, a novel Federated Retrieval-Augmented Generation system that integrates five key innovations to address the challenges of privacy, efficiency, and robustness in decentralized RAG. We have demonstrated the feasibility of our approach through a prototype implementation and provided a comprehensive framework for building practical FedRAG systems.

Our work opens up several avenues for future research. The immediate next step is to integrate a real Large Language Model and evaluate the end-to-end performance of the system on a downstream task, such as question answering. This would also allow for the implementation of a more realistic reward signal for the RLHF component. Further research could explore more advanced RL algorithms like PPO, investigate the trade-offs between different privacy-preserving techniques, and conduct a thorough security analysis of the system. Finally, deploying and evaluating FedRAG in a real-world setting with a large number of clients would be the ultimate test of its capabilities [15, 16, 17].

References

- [1] Yunfan Gao et al. "Retrieval-Augmented Generation for Large Language Models: A Survey". In: *arXiv preprint arXiv:2312.10997* (2024). DOI: 10.48550/arXiv.2312.10997.

- [2] Chaoyang He et al. “FedML: A Research Library and Benchmark for Federated Machine Learning”. In: *arXiv preprint arXiv:2007.13518* (2020). DOI: 10.48550/arXiv.2007.13518.
- [3] Tian Li et al. “Federated Learning: Challenges, Methods, and Future Directions”. In: *IEEE Signal Processing Magazine* 37.3 (May 2020), pp. 50–60. DOI: 10.1109/MSP.2020.2975749.
- [4] Nirupam Gupta, Shuo Liu, and Nitin Vaidya. “Byzantine Fault-Tolerant Distributed Machine Learning with Norm-Based Comparative Gradient Elimination”. In: *2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*. Taipei, Taiwan: IEEE, June 2021, pp. 175–181. DOI: 10.1109/DSN-W52860.2021.00037.
- [5] Martin Abadi et al. “Deep Learning with Differential Privacy”. In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS '16)*. 2016. DOI: 10.48550/arXiv.1607.00133.
- [6] Ian J. Goodfellow et al. “Generative Adversarial Networks”. In: *arXiv preprint arXiv:1406.2661* (2014). DOI: 10.48550/arXiv.1406.2661.
- [7] Kasper Green Larsen and Martin Ritzert. “Optimal Weak to Strong Learning”. In: *arXiv preprint arXiv:2206.01563* (2022). DOI: 10.48550/arXiv.2206.01563.
- [8] Peter Kairouz et al. “Advances and Open Problems in Federated Learning”. In: *arXiv preprint arXiv:1912.04977* (2021). DOI: 10.48550/arXiv.1912.04977.
- [9] Edward J. Hu et al. “LoRA: Low-Rank Adaptation of Large Language Models”. In: *arXiv preprint arXiv:2106.09685* (2021). DOI: 10.48550/arXiv.2106.09685.
- [10] Gautier Izacard et al. “Atlas: Few-shot Learning with Retrieval Augmented Language Models”. In: *arXiv preprint arXiv:2208.03299* (2022). DOI: 10.48550/arXiv.2208.03299.
- [11] Patrick Lewis et al. “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks”. In: *arXiv preprint arXiv:2005.11401* (2021). DOI: 10.48550/arXiv.2005.11401.
- [12] Huayang Li et al. “A Survey on Retrieval-Augmented Text Generation”. In: *arXiv preprint arXiv:2202.01110* (2022). DOI: 10.48550/arXiv.2202.01110.
- [13] H. Brendan McMahan et al. “Communication-Efficient Learning of Deep Networks from Decentralized Data”. In: *arXiv preprint arXiv:1602.05629* (2023). DOI: 10.48550/arXiv.1602.05629.
- [14] Virginia Smith et al. “Federated Multi-Task Learning”. In: *arXiv preprint arXiv:1705.10467* (2018). DOI: 10.48550/arXiv.1705.10467.
- [15] Yue Zhao et al. “Federated Learning with Non-IID Data”. In: *arXiv preprint arXiv:1806.00582* (2022). DOI: 10.48550/arXiv.1806.00582.
- [16] Parker Addison et al. “C-FedRAG: A Confidential Federated Retrieval-Augmented Generation System”. In: *arXiv preprint arXiv:2412.13163* (2024). DOI: 10.48550/arXiv.2412.13163.
- [17] Rachid Guerraoui et al. “Efficient Federated Search for Retrieval-Augmented Generation”. In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* (2025). DOI: 10.1145/3721146.3721942.

Behavioral Authentication using Password String Characteristics and Machine Learning Techniques *

Dylan Lott, Brandon Weathers , Yilian Zhang
Computer Science, Engineering & Mathematics
University of South Carolina Aiken
Aiken, SC 29801

{drlott, brw12, yilianz}@usca.edu

Abstract

Passwords and passkeys remain the most widely deployed authentication mechanisms to date, despite well-documented weaknesses regarding predictable modifications. Previous studies have shown that users often use strongly related passwords, creating predictable behavioral patterns that can be exploited for intrusion detection. In this paper, we focus on behavioral authentication using password strings alone. We evaluate the Electronic Personal Synthesis Behavior (EPSB) framework, the Longest Common substring (LCS) algorithm, and machine learning models using a large dataset constructed from public credential breaches. Our study shows that while individual algorithms have limitations, a combination of these algorithms with enriched features significantly improves performance. The strongest model achieved an accuracy of 89% with this hybrid approach.

1 Introduction

Empirical studies have consistently shown that users reuse passwords across multiple services or generate new passwords based on small systematic changes

*Copyright ©2026 by the Consortium for Computing Sciences in Colleges. Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the CCSC copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Consortium for Computing Sciences in Colleges. To copy otherwise, or to republish, requires a fee and/or specific permission.

to existing passwords. Users typically modify passwords by substituting substrings, changing capitalization, or appending digits and symbols. These predictable modifications produce clusters of related credentials that grow over time [1] [2].

These patterns promote research on authentication based on user behavior. The EPSB framework [3] is one approach that attempts to model user identity through statistical features, such as the mean and mode of password length or the count of capital letters and symbols. Although these types of approaches demonstrate effectiveness in detecting anomalous behavior, their dependence on basic statistics limits their ability to recognize real structural similarities between clusters of passwords. Recent research [4] also highlights the limitations of relying on isolated similarity metrics, such as Levenshtein distance and substring matching.

In this paper, we propose a novel hybrid approach. Rather than merely observing these similarity metrics as seen in prior work, we integrate them as structural features into a machine learning classifier. The password dataset, along with the outputs of the EPSB and LCS algorithms and these additional features, was used to train machine learning models. Three models were evaluated: Logistic Regression, Random Forest, and Gradient Boosting. Additionally, the most effective system was evaluated using XGBoost classifiers. This study shows that this integrated approach yields the strongest results. Furthermore, feature correlations were analyzed to assess effectiveness in detecting whether passwords were from the same user, as illustrated by the Heatmap in Figure 2.

2 Methodology

2.1 Overview

The current approach combines algorithmic analysis with machine learning methods to generate a binary classification determining whether two passwords originate from the same user. These methods were tested on a dataset constructed from public password leaks. The dataset is organized into positive and negative pairs to represent same-user and different-user scenarios. First, the raw password pairs are processed by the EPSB or LCS algorithms to produce intermediate statistical features and LCS ratios. Then, these outputs combined with additional character characteristics are fed into machine learning models to identify behavioral patterns in password creation.

2.2 ML Implementation

Three standard machine learning models were chosen to interpret the algorithmic outputs. The models chosen were: Logistic Regression, Random Forest, and Gradient Boost. Logistic Regression was chosen as a baseline model to evaluate the linear vs. non-linear nature of the data. The results from this

model varied across all sets and were unique compared to the other models, showing that the data often contains complex patterns. The other two classifier models were chosen as standards for this binary classification task. Due to the high performance of the final experiment, two additional XGBoost classifiers were run on this set; these two models were not run in any of the other experiments.

These models were all trained on the same dataset with a 75-25 train-test split. Feature sets varied based on which algorithmic approach was used. In previous similar experiments [3], machine learning algorithms were not used, and no standard of accuracy was presented. This previous study suggested the use of the LCS algorithm in place of EPSB, but this study shows that even a basic implementation of the EPSB algorithm is significantly better than the LCS algorithm in isolation.

2.3 EPSB Implementation with ML Interpretation

EPSB algorithm was initially evaluated in isolation to determine the predictive strength of basic statistical properties regarding passwords. The implementation conducted in this study extracted per-password distributions of capital letters, lowercase letters, combined capital and lower case letters, digits, certain symbols, and length. Prior studies and researcher expectations implied that EPSB in isolation would not provide strong discriminatory power [3]. The algorithm is designed to parse a set of passwords associated with a single user to create a statistical profile for the user. New passwords are then checked against the profile to determine whether they are from the legitimate user. In this study, these derived statistical features were combined with the raw data set to construct a new dataset for the machine learning models.

2.4 LCS Implementation with ML Interpretation

The initial LCS approach compares password pairs to produce an LCS-ratio which equals the length of the longest common substring divided by the average length of both passwords. This produces a comprehensive one-dimensional statistic to quantify the predictive power of a substring. The standard suffix tree method is used to implement the LCS algorithm. This method can be applied recursively, returning the longest string, then the second longest string, then the third longest. We have limited this algorithm to only return substrings that are longer than two letters and to recursively execute a maximum of three times. The K-shaped graph derived from the LCS algorithm implies that when passwords are similar, they are highly similar, emphasizing the importance of retaining this feature. These observations motivated the inclusion of additional structural and semantic similarity measurements.

2.5 LCS/EPsB Implementation with Annotations & ML Interpretation

To address limitations within an isolated LCS-based or EPsB-based approach, new models were developed to enrich the predictive power of the dataset. This model introduced several key features that significantly increased accuracy, including password lengths, length difference, shared character count, the Levenshtein distance of the two strings, and the Jaccard similarity coefficient of the two strings [5]. As shown in Figure 3 and Figure 4, Jaccard similarity and Levenshtein distance are among the most important predictive features. These figures also suggest that the longest common substring is highly predictive.

2.6 LCS for Multiple Sub-Strings with EPsB annotations & ML Interpretation

The final model implementation integrates a modified version of the LCS algorithm designed to retrieve the three longest common substrings. The ratios for the second and third longest substrings to the total length of the string (LCS 2 ratio and LCS 3 ratio) were added to the dataset. These new metrics were combined with the EPsB features and enriched features discussed above to form a comprehensive feature set. This approach proved to be the most effective approach that captures complex behavior pattern well with little increase in computational cost.

3 Experiment

3.1 Dataset Selection and Issues

A dataset of 272,000 password pairs was constructed from two public breach sources containing plain text credentials. These included a breach of a gaming platform and a breach of a widely accessed entertainment platform. Both breaches contained several million records. To maintain ethical standards, all email identifiers were hashed and location fields were discarded. Only password fields were retained, with no other identifying information. This dataset consisted of 136,000 positive pairs, defined as passwords associated with the same hashed email, reflecting behavioral consistency from a single user. Using this practice of matching emails yielded the first set of 136,000 pairs. To create a balanced dataset, an additional 136,000 negative pairs were derived from the same two data breaches by randomly selecting unique email identifiers to create pairs.

Five distinct feature sets were derived to train Machine Learning Models. Table 1 shows the details of the datasets.

3.2 Experiment Metrics

The performance of the models was evaluated using standard binary classification metrics: accuracy, precision, recall, and F-1 Score. In this context, a positive classification represents a pair of passwords from the same individ-

Table 1: Composition of Feature Sets

Dataset	Methodology	Features Included
Set 1	EPSB	Password pairs, capital letter count, lowercase count, total letter count, digit count, symbol count.
Set 2	EPSB & Enriched Features	features above and Levenshtein distance, Jaccard similarity, shared character count, length difference.
Set 3	Basic LCS	Password pairs, LCS ratio, LCS length.
Set 4	LCS & Enriched Features	Password pairs, multiple LCS ratios (1, 2, & 3), shared character count, Levenshtein distance, Jaccard similarity, length difference.
Set 5	LCS, EPSB & Enriched Features	All features from Set 2 and Set 4 combined.

ual, while a negative classification represents pairs from different individuals. The precision scores reflect the model’s ability to avoid incorrectly classifying negative pairs, while recall measures its ability to correctly identify positive pairs. The F-1 score summarizes these competing objectives and is the most comprehensive measurement of effectiveness for behavioral authentication.

All experiments were conducted using the same standard dataset with varying features included. 75% of the sample data is allocated for training and the rest of 25% data is reserved for testing model accuracy.

4 Results

A comparative summary of the F-1 score results from the four methodologies is presented in Figure 1.

4.1 EPSB and EPSB with Enriched Features

When evaluated in isolation, EPSB-derived statistical features showed gaps between linear and tree-based models. Tree-based models such as Random Forest and Gradient Boosting models achieved around 85% accuracy, while Logistic Regression performed notably worse. This suggests that nonlinear relations between string composition features capture meaningful behavioral patterns. And the interesting observation is that, once enriched features such as Levenshtein distance, Jaccard similarity were added to the dataset, the accuracy improved significantly across all models (ranging from 81% to 88%). This demonstrates the combination of EPSB statistics and enriched features

successfully capture critical factors of password behavioral patterns.

4.2 LCS

The standalone LCS algorithm yielded the lowest performance. This model also experienced a noticeable imbalance between class recalls, achieving a 0.88 recall for true negatives, but only a 0.58 recall for matching pairs. This imbalance reinforces the fact that passwords from the same individual do not always have a long common substring.

4.3 LCS with Enriched Features

Incorporating string-level features such as Levenshtein distance greatly improved the accuracy across all classifiers over the basic LCS model, implying that these additional metrics capture global similarity as well as localized edits and structural characteristics that are not reflected by the LCS algorithm. The consistency of these metrics across linear models and classifiers also suggests that the new dataset provides a more linearly identifiable representation of behavioral similarity.

4.4 LCS with Enriched & EPSB Features

The comprehensive hybrid approach yielded the strongest overall performance across all evaluated models. By combining the statistical composition(EPSB) with structural similarity (LCS & Enriched), the system achieved the highest accuracy of 88.2% with Gradient Boosting Model. Two additional models were run on this set, consisting of two variations of XGBoost classifiers which achieved the study’s highest accuracy of 89%.

System	Highest Accuracy	Logistic Regression	Random Forest	Gradient Boost
EPSB	85.4%	76.4%	85.4%	84.7%
EPSB & Enriched Features	88%	81.4%	87%	88%
LCS	76.0%	73.0%	76.0%	75.9%
LCS & Enriched Features	80.7%	79.9%	79.6%	80.7%
LCS, EPSB, & Enriched Features	88.2%	81.8%	87.3%	88.2%

Figure 1: Highest Accuracy Scores Across Systems

4.5 Correlation of Features

A correlation feature heatmap (Figure 2) has been created to analyze relationships between different features. As expected, strong positive correlations exist between length-based metrics (Len1, Len2) and character counts (letters1, numbers1). Interestingly, Levenshtein distance shows a strong negative

correlation with LCS-based metrics and Jaccard similarity, confirming that as distance increases, structural similarity decreases.

The significance analysis of characteristics is illustrated in Figure 3 and Figure 4. Both results repeatedly emphasize the importance of Jaccard Similarity, Levenshtein Distance, and LCS-ratio for the first common substring. However, features derived from the second and third longest common substrings were identified as less important.

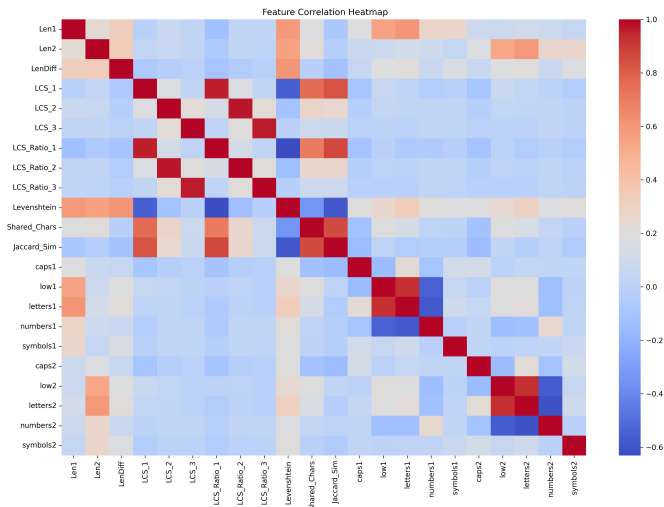


Figure 2: Correlation of Features

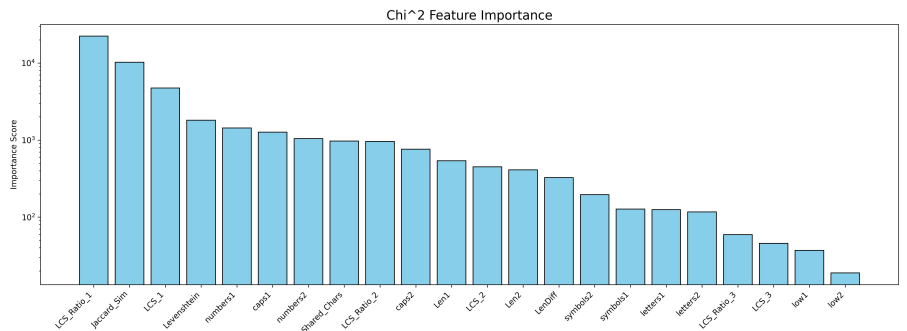


Figure 3: CHI² Score

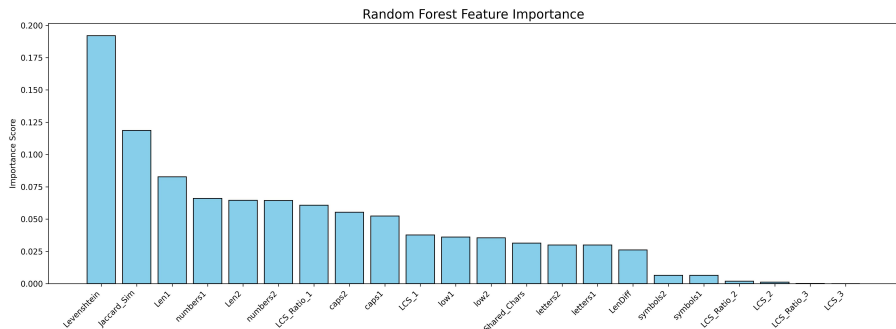


Figure 4: Random Forest Feature importance

5 Future Work and Discussion

This study demonstrates that behavioral authentication is possible and practical through password string characteristics alone. These highly accurate results were achieved without relying on temporal, contextual, or biometric data. This paper proposed a new hybrid approach that combined the LCS algorithm, EPSB-style statistics, and enriched string metrics, achieving a peak accuracy of 89% using XGBoost classifiers. An interesting observation of this research is the role of feature types in model performance. While statistical composition features (EPSB) primarily capture non-linear behavioral patterns effective in tree-based models, the inclusion of structural metrics—such as Jaccard similarity and Levenshtein distance—significantly improved linear separability.

Compared with other machine learning methods, our algorithmic approach is lightweight, making it easier to use in real-time authentication. When a user attempts to log in or update a password, the system evaluates the input against their established behavioral profile. If the pattern varies greatly, the system can trigger alerts or require further authentication. This behavioral analysis functions as an intermediate step that runs before the password is hashed. It is also important to address the security concerns associated with these algorithms. Traditionally, the LCSM class of algorithms requires storing plaintext passwords in memory, introducing a potential vulnerability. However, in this suffix tree implementation, passwords are not persistently stored. Instead, the algorithm processes each input string by breaking it into characters and structuring them within a specialized Trie. Once the Trie is searched to find the longest common substring, it is immediately discarded. Because this entire process occurs transiently in memory, the system avoids storing plaintext passwords in any persistent database.

In this paper, standard classifiers were selected to establish a reliable baseline due to their success in similar research. The model is stable across multiple runs. Optimization techniques were not used in the current approach; how-

ever, further investigation will be done in this area. Although legitimate and verifiable breach datasets are exceptionally difficult to find, future research will test these algorithms on cross-dataset generalization.

Acknowledgment

The authors wish to thank Dr. Rao Li at University of South Carolian Aiken for his guidance in this project, and reviewers for their valuable suggestions.

References

- [1] Dinei Florêncio and Cormac Herley. “A Large-Scale Study of Web Password Habits”. In: *Proceedings of the 16th International World Wide Web Conference (WWW)*. 2007, pp. 657–666.
- [2] Sacha Gaw and Edward W. Felten. “Password Management Strategies for Online Accounts”. In: *Proceedings of the 2nd Symposium on Usable Privacy and Security (SOUPS)*. 2006, pp. 44–55.
- [3] Boumedyen Shannaq and Mohanaad Shakir. “Enhancing Security through Multi-Factor User Behavior Identification: Moving Beyond the Use of the Longest Common Subsequence (LCS)”. In: *Informatica* 48.19 (2024), pp. 57–72. DOI: 10.31449/inf.v48i19.6270.
- [4] Davide Berardi et al. “Password Similarity Using Probabilistic Data Structures”. In: *Journal of Cybersecurity and Privacy* 1.1 (2021), pp. 78–92. ISSN: 2624-800X. DOI: 10.3390/jcp1010005. URL: <https://www.mdpi.com/2624-800X/1/1/5>.
- [5] Sam Fletcher and Md Zahidul Islam. “Comparing sets of patterns with the Jaccard index”. In: *Australasian Journal of Information Systems* 22 (2018).

Learning Cryptography One Tap at a Time: An NFC-Based Payment System for the Classroom*

Bilal Shebaro
Department of Computer Sciences
St. Edward's University
Austin, TX 78704
`bshebaro@stedwards.edu`

Abstract

Cryptography is a foundational topic in computer science education and is commonly introduced through algorithmic and mathematical frameworks. Building on this foundation, students can benefit from complementary learning experiences that illustrate how these principles operate within real-world systems. This paper presents a context-first approach that situates cryptographic concepts within a practical and tangible workflow, an NFC-based digital payment system, allowing students to connect theoretical knowledge with applied system behavior.

We introduce a closed-loop, mock digital payment system that operates entirely on synthetic data, in which students design a secure NFC-enabled payment system while interacting with services that simulate tokenization, key management, and transaction authorization. The approach enables student discussions in system-level reasoning, threat modeling and attacks, and correct application of cryptographic primitives. Through hands-on implementation, simulations, and adversarial demonstrations, students gain deeper insight into how cryptography enforces security guarantees in modern systems. By reframing cryptography education around systems that students already trust and use, this approach seeks to bridge the gap between theory and practice, fostering deeper understanding of how cryptographic mechanisms enable secure interactions in modern computing environments.

*Copyright ©2026 by the Consortium for Computing Sciences in Colleges. Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the CCSC copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Consortium for Computing Sciences in Colleges. To copy otherwise, or to republish, requires a fee and/or specific permission.

1 Introduction

Cryptography is a foundational component of modern computing systems, underpinning secure communication, authentication, data integrity, and digital payments. Traditional cryptography instruction is often introduced through abstract algorithms and mathematical models that form a solid theoretical base, and students often benefit from opportunities to see these concepts applied within concrete system contexts. As a result, these opportunities can further support students in developing intuition about why cryptographic mechanisms are needed, how they are applied in practice, and what security guarantees they provide [1, 2, 3, 4].

At the same time, students routinely interact with cryptography-enabled systems in their daily lives. Contactless payment technologies, such as mobile wallets and tap-to-pay transactions, are widely adopted and trusted by users. While students rely on these systems for convenience and security, they rarely understand the cryptographic processes that enable them. Concepts such as tokenization, key management, message integrity, replay protection, and authentication are largely invisible to users, creating a disconnect between everyday technological trust and formal cryptographic understanding [5, 2, 6, 7].

This paper argues that this disconnect presents a powerful educational opportunity [8, 9]. Rather than introducing cryptography through abstract algorithms alone, we propose a context-first pedagogical approach that introduces cryptographic concepts in a familiar and tangible workflow through a contactless digital payment system. We present a hands-on instructional module centered on a mock, Apple Pay-inspired Near-Field Communication (NFC) payment system [10], designed specifically for classroom use. In this closed-loop environment, students design and implement a digital payment application that performs secure NFC-based transactions using synthetic payment data, while interacting with instructor-provided backend services that simulate key components of a real payment ecosystem.

The mock system mirrors essential elements of modern mobile payment architectures, such as tokenization (primary account number to device account number conversion), cryptographic payload generation, replay protection, and backend authorization, without interfacing with real financial networks or sensitive data [11]. By situating cryptographic primitives within a concrete payment workflow, students encounter cryptography as a necessary response to identifiable threats rather than as an isolated mathematical exercise [12].

The remainder of this paper is organized as follows: Section 2 describes the context-first pedagogical approach, including system framing and active and adversarial learning. Section 3 describes the architecture and design of the mock digital payment system. Section 4 describes the classroom activities,

demonstrations, and learning outcomes, followed by the paper conclusion in Section 5.

2 Pedagogical Design

The instructional design of the NFC-based secure payment module follows a context-first approach to cryptography, in which system behavior and security goals precede the introduction of cryptographic primitives. Rather than beginning with mathematical definitions or algorithmic properties, students are first exposed to a realistic payment scenario and are asked to reason about what could go wrong if appropriate protections are not in place. Cryptographic concepts are then introduced incrementally as necessary responses to specific threats.

2.1 Framing the Payment System

Instruction begins with a high-level walkthrough of a contactless payment interaction from a user's perspective: an NFC device taps a terminal, a transaction is processed, and an approval or rejection is returned. Students are encouraged to identify implicit assumptions in this workflow, such as trust in the NFC device information, the terminal, the network, and backend services.

Students are then asked to identify the high-level components required to support a digital payment system, such as the NFC device (or mobile wallet), a merchant terminal, and backend services, focusing on their roles rather than implementation details.

The discussion then follows with an exploration of what it means for a payment system to be secure from a user and system perspective. Students are prompted to articulate expected properties, such as successful authorization, resistance to fraud, protection of sensitive data, and usability. Particular emphasis is placed on the expectation that real payment systems do not store or transmit raw credit card numbers, motivating early discussion of tokenization and data minimization without yet introducing cryptographic details.

The instructor then guides students through a thought experiment in which these components communicate using plaintext messages. Students are asked to consider what information would be exchanged, who can observe or modify that communication, and what could go wrong if no protections are in place. This discussion naturally surfaces concerns such as message tampering, impersonation, replay, and unauthorized reuse of credentials.

Only after these risks are discussed, it becomes clear to students that different forms of cryptography are needed for securing such a payment system. Students propose requirements such as ensuring messages are not modified, verifying the sender of a transaction, preventing reuse of old payment requests, and limiting exposure of sensitive identifiers. These requirements serve as a bridge between system goals and cryptographic mechanisms, preparing students to understand why integrity protection, authentication, freshness, and

tokenization are necessary.

2.2 Active and Adversarial Learning

A central component of the pedagogical design is the inclusion of adversarial exercises. Students are required not only to implement secure transactions, but also to demonstrate how their systems respond to intentional misuse. Typical adversarial scenarios include replaying previously valid transactions, modifying transaction amounts, or attempting to use revoked tokens. Each scenario motivates the introduction of a corresponding cryptographic mechanism. For example, integrity violations motivate digital signatures or message authentication codes, while replay attacks motivate nonces or timestamps. In this way, cryptographic primitives are framed as solutions to concrete problems, not abstract requirements.

3 Digital Payment System Overview

This section describes the architecture and design of the mock digital payment system used in the instructional module. The system is inspired by modern mobile payment platforms such as Apple Pay, but is intentionally simplified and implemented as a closed-loop educational simulator for the purpose of exposing students to the essential cryptographic and system-level components that enable secure contactless payments.

3.1 High-Level Architecture

At a high level, the system consists of five logical components:

1. Enrollment Device (ED) / Payment Device (PD): NFC tag-based Device Terminal and Reader.
2. Merchant Bank Server (MB): acting as the recipient of the transferred money, also known as the Acquirer.
3. Issuer Bank Server (IS): acting as the sender of the transferred money. In a real-world scenario, this is the financial institution (e.g.bank) that provides credit, debit, or prepaid cards.
4. Token Service Provider (TSP): responsible for storing and matching Personal Account Numbers (PANs) and Device Account Numbers (DANs). Tokens usually stored in a Token Vault.
5. Key Management Service (KMS): responsible for issuing keys (public-private pair, or symmetric) for the different system entities.

The NFC tag communicates with the payment reader terminal via a short-range NFC interaction and with backend services (TSP, IS, MB) via standard HTTPS APIs. All sensitive operations are performed using synthetic identifiers and cryptographic keys generated by the KMS are specifically for instructional use.

3.2 Enrolling Phase: Mock Wallet Enrollment

The system begins with an enrolling phase, analogous to adding a payment card to a mobile wallet. In our instructional design, the NFC tag is acting like the mobile wallet card under enrollment. A mock primary account number (PAN) is generated using synthetic data and is stored in the NFC tag. This PAN is never used directly in transactions. Instead, the wallet communicates with the instructor-provided Token Service Provider (TSP), which converts the PAN into a device account number (DAN) through a tokenization process. The TSP maintains a secure mapping between PANs and DANs and tracks token lifecycle state (e.g., active, revoked).

In this phase, students examine an intentionally insecure enrollment process and identify the security risks involved. Figure 1 illustrates an insecure payment card enrollment process in which sensitive card information is transmitted and stored without appropriate cryptographic protections. During enrollment, the enrollment device sends raw card data, the PAN, to backend services and is converted to the DAN without encryption or secure authentication, exposing the system to risks such as data interception, unauthorized reuse, and long-term credential compromise.

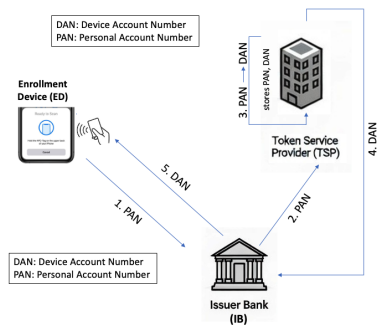


Figure 1: Simplified example of an insecure payment card enrollment process in which raw card data is transmitted without sufficient cryptographic protection.

After discussion of risks and attacks possible, the discussion should lead to a more secure enrollment phase as illustrated in Figure 2, establishing a trusted foundation for subsequent transactions. During such discussion, students realize the need for encryption and signatures, thus the need for a Key Management Service to issue cryptographic keys to associate with every enrollment instance. Public keys required for verification are made available to backend services, while private keys remain on each system component within our simplified trust model.

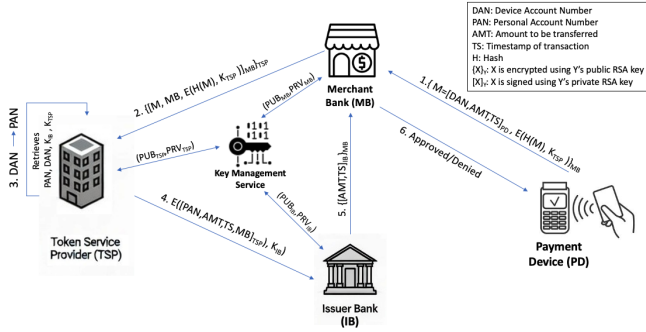


Figure 4: Simplified secure payment card transaction process using proper cryptographic authentication to protect sensitive card data.

the necessary transaction information such as merchant identifier, terminal identifier, a freshness value (e.g., a nonce or timestamp), and finally the transaction amount along with the PAN stored on the tapped NFC tag.

The payment system applies cryptographic protections to the payload, including hashing, encryption and digital signatures, and transmits it to the backend system for authorization. A log system is provided to the students to display the status of the information transfer from one backend component to another.

The submission of the payload transaction results in a success (approved) or failure (denied) response. This is managed through backend services that simulate the roles of a issuer bank server (sender of the money), merchant bank server (receiver of the money), and the token service provider. When a transaction request is received, a series of validation steps are performed, including cryptographic verification of the transaction payload, replay protection checks, and token status verification through the TSP. If all checks are successful, the transaction is approved, and a response is returned to the payment device application. If not, a structured decline response is issued, allowing students to observe and diagnose failure cases.

4 Classroom Activities and Demonstrations

The NFC-based secure payment module is structured around a series of hands-on classroom activities that culminate in live demonstrations of secure and adversarial transactions. Students examine how a contactless payment progresses from a physical NFC tap to backend authorization and approval, all through logs that are made visible for each system component. During this walkthrough, emphasis is placed on identifying trust boundaries, data flows, and potential points of failure. Students are asked to predict what information must be protected and what threats may arise at each stage of the transaction. This primes students to view cryptography as a response to system-level risks

rather than as an abstract requirement.

4.1 Secure vs. Adversarial Demonstrations

A core feature of the module is a live tap-to-pay demonstration. Students demonstrate a successful secure transaction (using synthetic data) by tapping the NFC terminal reader against an NFC tag and receiving an authorization response in real-time.

During the demonstration, students are required to explain:

- What data is exchanged during the tap
- Which cryptographic protections are applied
- Why the transaction is accepted or declined

For successful transactions, students must demonstrate at least one intentional attack scenario and explain why it fails. Common adversarial demonstrations include:

- Replaying a previously valid transaction payload
- Modifying transaction amounts or metadata
- Attempting transactions with revoked tokens

4.2 Observed Learning Outcomes

This module was deployed in an introductory cybersecurity upper-level undergraduate computer science course with approximately 40 students. Observations and discussions were collected through structured live demonstrations of secure and adversarial payment scenarios where several consistent patterns emerged. Students demonstrated a stronger ability to articulate the purpose of cryptographic mechanisms in the context of system behavior, rather than describing them solely in algorithmic terms. Post-module responses more frequently referenced specific threats (e.g., replay attacks, tampering) and corresponding defenses, suggesting improved security reasoning in all other course topics.

During live demonstrations, students were generally able to explain not only why successful transactions were accepted, but also why adversarial attempts were rejected. This ability to reason about failure cases signifies a shift from a surface-level understanding of cryptographic concepts to a profound comprehension and practical implementation of cryptographic functionalities and applications.

5 Conclusion

This paper presents a context-first instructional approach to cryptography that complements traditional foundations by situating cryptographic concepts within a hands-on, NFC-based mock digital payment system inspired by contemporary mobile wallets. Through live demonstrations, adversarial exercises, and explicit justification of design decisions, students engage with cryptography as a system-enabling tool rather than an abstract collection of algorithms.

Core security properties, such as authenticity, integrity, confidentiality, replay protection, and tokenization, are introduced as responses to identifiable threats within a familiar payment workflow.

By emphasizing correct cryptographic usage, protocol design, and security reasoning through structured discussion and live demonstrations, the approach supports deeper conceptual understanding of how cryptography operates within real systems. The NFC-based payment module presented in this work provides a replicable and adaptable educational framework for bridging theory and practice in applied cryptography instruction and for enriching undergraduate computing curricula with meaningful, system-oriented learning experiences.

References

- [1] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography*. 3rd. CRC Press, 2020.
- [2] Ross Anderson. *Security Engineering: A Guide to Building Dependable Distributed Systems*. 3rd. Wiley, 2020.
- [3] Matt Bishop. “Education in Information Security”. In: *IEEE Concurrency* 8.4 (2000), pp. 4–8.
- [4] Dieter Gollmann. *Computer Security*. eng. 3rd ed. Chichester, West Sussex, U.K. ; Hoboken, N.J.: Wiley, 2011. ISBN: 9780470741153.
- [5] EMVCo. *EMV Payment Tokenisation Specification: Technical Framework*. EMVCo. Overview document. 2014.
- [6] Bruce Schneier. *Secrets and Lies: Digital Security in a Networked World*. Wiley, 2000.
- [7] Paul A. Grassi et al. *Digital Identity Guidelines*. Tech. rep. SP 800-63-4. Supersedes NIST SP 800-63-3. National Institute of Standards and Technology, Aug. 2025. DOI: 10.6028/NIST.SP.800-63-4. URL: <https://www.nist.gov/publications/nist-sp-800-63-4-digital-identity-guidelines>.
- [8] David A. Kolb. *Experiential Learning: Experience as the Source of Learning and Development*. Prentice Hall, 1984.
- [9] National Research Council. *How People Learn: Brain, Mind, Experience, and School: Expanded Edition*. Washington, DC: The National Academies Press, 2000. ISBN: 978-0-309-07036-2. DOI: 10.17226/9853. URL: <https://nap.nationalacademies.org/catalog/9853/how-people-learn-brain-mind-experience-and-school-expanded-edition>.
- [10] Roy Want. “Near Field Communication”. In: *IEEE Pervasive Computing* 10.3 (2011), pp. 4–7.
- [11] Waqas Ahmed et al. “Security in Next Generation Mobile Payment Systems A Comprehensive Survey”. In: *IEEE Access* 9 (Aug. 2021). DOI: 10.1109/ACCESS.2021.3105450.
- [12] Chris Dede. “Immersive Interfaces for Engagement and Learning”. In: *Science* 323.5910 (2009), pp. 33–35. DOI: 10.1126/science.1167311. URL: <https://www.science.org/doi/abs/10.1126/science.1167311>.

Comparison of Software Development and Communication Skills by Degree Options *

Karern Works¹, Sharanya Jayaraman²

¹ Computer Science

Florida State University Panama City

Panama City, FL 32405

² Computer Science

Florida State University

Tallahassee, FL 32306

{keworks, sjayaraman}@fsu.edu

Abstract

Employers in the Computer science Industry, as in most STEM disciplines, want prospective Computer Science graduates to possess strong software development and technical communication skills. Computer science has long included specialized disciplines such as cybersecurity, computational biology, data science, and AI. Consequently, many institutions offer degree programs that integrate traditional computer science with these specialized areas. In addition, these institutions often also offer Bachelors of Arts and Science degrees in traditional computer science. Given the current technical job market, it is imperative that institutions ensure that the proficiencies obtained by all their graduates match the skills required by employers. In this study, we analyze the software development and communication skills of 1) employer requirements and 2) undergraduate proficiencies across different degree options. Our findings indicate that job postings for computer science sub-majors reference software engineering skills less frequently and oral communication skills more frequently than postings for traditional computer science majors.

*Copyright ©2026 by the Consortium for Computing Sciences in Colleges. Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the CCSC copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Consortium for Computing Sciences in Colleges. To copy otherwise, or to republish, requires a fee and/or specific permission.

1 Introduction

The rapid evolution of the technology sector has increased employer expectations that computer science graduates possess not only strong software development abilities but also effective technical communication skills[1]. As the discipline has expanded to include specialized areas such as cybersecurity, data science, computational biology, and artificial intelligence, universities have responded by offering a range of degree programs that blend traditional computer science with these emerging fields[2]. Alongside these specialized pathways, institutions continue to offer traditional Bachelor of Arts and Bachelor of Science degrees in computer science. However, as the technical job market becomes increasingly competitive and diversified, questions remain about whether the skills developed across these degree options align with employer needs. This study investigates employer requirements and undergraduate proficiencies in software development and communication skills across different computer science degree pathways to better understand how well current programs prepare graduates for the workforce.

2 Related Work

2.1 Perceived preparedness and self-efficacy at graduation

A common approach to understanding graduates' readiness is to measure self-perceived preparedness, often framed through self-efficacy theory—i.e., an individual's belief in their ability to succeed at specific tasks. Bandura's foundational work positions self-efficacy as a strong predictor of motivation, persistence, and performance, while also noting that perceptions can diverge from objective skill levels depending on experience and feedback [3].

In higher education assessment, exit and alumni surveys are frequently used to capture how confident students feel about program outcomes at the point of graduation (or shortly after), providing a complementary lens to direct assessment artifacts. ABET's accreditation ecosystem, for example, explicitly accommodates indirect measures such as surveys as part of a broader evidence base for student outcomes and program educational objectives[4].

2.2 Program outcomes, accreditation outcomes, and competency frameworks

Studies that compare graduates' perceptions to external expectations typically need a shared vocabulary for "capabilities." Many programs operationalize capability using (a) accreditation-driven outcomes, and/or (b) widely adopted competency frameworks. ABET's outcomes-based model encourages programs to define measurable student outcomes and gather evidence through a combination of direct and indirect methods[5].

Within computing specifically, the Computer Science Curricula 2023 (CS2023) guidelines synthesize current perspectives on what CS graduates should know

and be able to do, including attention to evolving platforms and professional practice expectations shaped by industry[6].

In parallel, cross-disciplinary workforce frameworks such as NACE Career Readiness Competencies provide a standardized set of professional competencies (e.g., communication, teamwork, critical thinking, professionalism) that are frequently used in employer-facing discussions of “job readiness,” and can be mapped to survey items or learning outcomes.

2.3 Evidence of perception gaps between graduates and employers

A recurring theme in the broader employability literature is that students tend to rate their proficiency higher than employers do, particularly for “durable” or professional skills. NACE reports quantify these gaps across multiple competencies, highlighting substantial differences in perceived proficiency between new graduates and employers (notably leadership/professionalism, critical thinking, and communication).

Earlier work also examined mismatches between student perceptions and employer expectations across job-relevant skills, supporting the idea that perception data is informative but should be interpreted alongside external benchmarks[7].

Within computing and IT-adjacent disciplines, multiple studies report that employers emphasize not only baseline technical ability but also professional skills—communication, teamwork, problem-solving, and work habits—and that gaps are frequently observed in these areas

2.4 The academia–industry gap in computing and software fields

Computing education research and industry-facing studies often characterize an “academia–industry gap” as a mismatch between what programs emphasize and what entry-level roles demand, especially around professional practice and applied experience. Recent work in the ACM community continues to document these gaps and frames them as persistent and multi-factorial (curricular constraints, rapidly changing tools, differences in incentives, etc.)[8].

Related research on “work readiness” similarly treats readiness as multi-dimensional—technical competence plus professional attributes and context-specific application—and argues that measurement should include both graduate self-perception and employer-side evaluation[9].

2.5 Using exit/alumni surveys and indirect measures in outcomes assessment

Exit surveys are widely used as indirect assessment inputs in accreditation and continuous improvement workflows because they can be administered at scale and can capture constructs that are difficult to measure via exams alone (confidence, perceived growth, perceived relevance). ABET-oriented assessment examples explicitly include alumni or graduate surveys asking respondents to rate preparedness and the relevance of program educational objectives to their

careers, sometimes with structured outreach mechanisms to reach alumni cohorts.

Methodological work in outcomes assessment emphasizes that survey-based data is most defensible when (1) items are aligned to clearly defined outcomes/competencies, (2) results are triangulated with direct measures, and (3) the program closes the loop by using findings to guide curricular decisions[10].

2.6 Comparing student perceptions to “industry requirements”

To compare exit-survey perceptions to industry needs, prior work typically operationalizes “industry requirements” through one or more proxies: employer surveys/interviews, advisory board input, competency frameworks, or analyses of job postings. Studies focused on employability and skills mismatch commonly treat job readiness as alignment between graduate capabilities (perceived or demonstrated) and employer expectations (stated or observed), often finding that professional skills and applied experience are the most sensitive mismatch areas.

3 Data Collection

3.1 Employer Requirements

Table 1: Sample of Job Degree Common Terms

Degree	Terms
Subfield	"computational biology", "bioinformatics" "cybersecurity", "information security analyst"
BA-CS	"software developer", "web developer" "database administrator", "technical support engineer"
BS-CS	"software engineer", "devops engineer" "systems architect", "data engineer"

To investigate employer requirements, we analyzed 123,849 LinkedIn job ads posted between 2023 and 2024. The job descriptions were preprocessed by converting text to lowercase, removing common stop words[11], and standardizing synonyms to a single term[12].

From the cleansed dataset, we focused on entry-level positions requiring a bachelor’s degree by selecting ads containing phrases such as “entry level” or “bachelor of [xxx] degree required.” Within this subset, we identified 2,615 ads requiring a bachelor’s degree computer science subfield, 1,310 ads requiring a Bachelor of Arts in traditional computer science, and 5,567 ads requiring a Bachelor of Science in traditional computer science by further analyzing for terms commonly associated with each degree type. Table1 presents a sample of common terms associated with each degree type. For the remainder of this study, we refer to these categories as subfield, BA-CS, and BS-CS.

3.2 Undergraduate Proficiencies

To address our question about undergraduate proficiencies, we analyzed Likert-scale exit survey responses from 132 undergraduates collected between 2021 and 2025. The survey included questions about students’ perceptions of their own software development and communication proficiencies. Our dataset included students from two Bachelor of Arts majors—Computer Programming and Applications and Computer Science—and three Bachelor of Science majors: Cyber Criminology, Computational Biology, and Computer Science. Respondents self-reported the degree they received. For analysis, the data were divided into three groups: 88 Computer Science Bachelor of Science majors; 35 Computer Science Bachelor of Arts majors (Computer Programming and Applications and Computer Science combined); and 9 Computer Science subfield majors (Cyber Criminology and Computational Biology combined).

4 Data Analysis

4.1 Employer Requirements

We begin by analyzing the LinkedIn job advertisement dataset described earlier, focusing on ads containing terms related to software engineering, oral communication, written communication, and technical documentation skills. Table 2 provides a sample of common terms associated with each skill category. The primary metric in this analysis is the percentage of job ads that include at least one phrase related to a given skill (%).

Table 2: Sample of Job Ad Skill Common Terms

Skill	Terms
Software Engineering	"develop applications", "write software" "build applications", "construct codebases"
Oral Communication	"verbal communication skills", "oral communication" "spoken communication", "public speaking"
Written Communication	"written communication skills", "writing abilities" "skilled in writing", "proficient in business writing"
Technical Documentation	"technical documentation", "software documentation" "system documentation", "engineering documentation"

As expected, the results show that job advertisements mentioning software engineering skills are most common for traditional computer science degrees, with BS-CS at 41.1% and BA-CS at 34.7% (see Table 3). In contrast, only 9.8% of job advertisements in subfields mention software engineering skills.

As shown in Table 4, written communication skills are referenced far more frequently than oral communication skills. References to written communica-

Table 3: Job Ads Software Engineering

Metric	Subfield	BA-CS	BS-CS
%	9.8	34.7	41.1

tion occur at similar rates across all groups: subfield positions (92.8%), BA-CS (91.9%), and BS-CS (91.9%). In contrast, oral communication skills are mentioned more often in subfield job advertisements than in traditional computer science roles, with frequencies of 19.5% for subfield positions, 9.8% for BA-CS, and 10.8% for BS-CS. As shown in Table 5, technical documentation skills appear at similar rates for BA-CS (9.0%) and BS-CS (8.1%) job ads, while subfield positions are slightly lower at 6.19%.

Table 4: Job Ads Communication

Metric	Subfield	BA-CS	BS-CS
Oral %	19.5	9.8	10.8
Written %	92.8	91.9	91.0

Table 5: Job Ads Technical Documentation

Metric	Subfield	BA-CS	BS-CS
%	6.19	9.0	8.1

4.2 Undergraduate Proficiencies

We now present an analysis of the aforementioned exit survey results. All questions used the following 5-point Likert scale: Strongly Agree (5), Agree (4), No Opinion/Not Applicable (3), Disagree (2), and Strongly Disagree (1). The frequency distribution of responses across the subgroups serves as a basis for comparison analysis.

There was one question that asked the respondents to evaluate their software engineering proficiencies: "I can design and implement software solutions". Per the frequency distribution chart on software engineering (See Figure 1) , the BS Computer Science reported the highest proficiency with 95.4% strongly agreeing or agreeing. BA Computer Science students followed closely at 91.43%. In contrast, only 77.8% of the Computer Science Subfield students selected agree, no students selected strongly agree.

There were two questions on technical communication skills. The first evaluated technical documentation skills: "I can completely document programs". The second assessed technical oral presentation skills: "I can make clear and informative oral presentations on programs".

As shown in Figure 2), similar to software engineering proficiencies, both

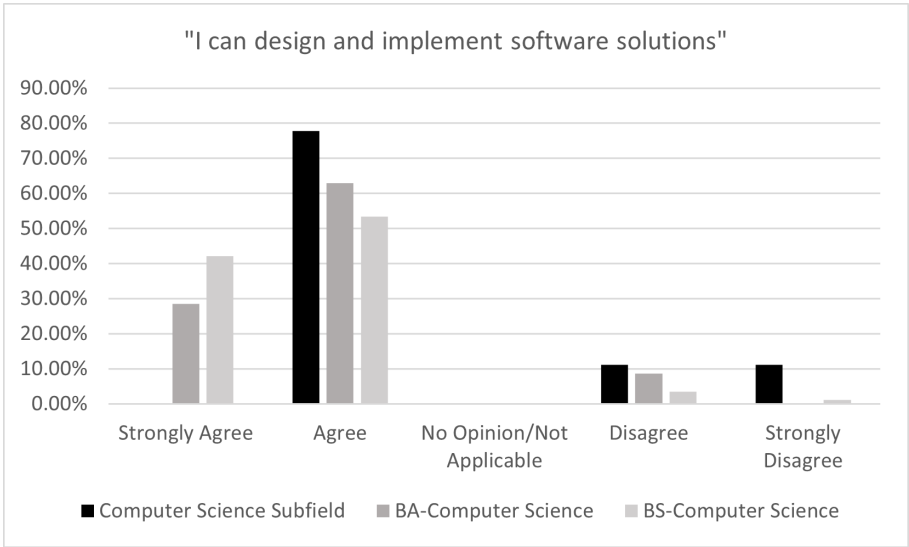


Figure 1: Frequency Distribution: Software Engineering

BS and BA Computer Science students reported the high proficiency in technical documentation, with 90.9% and 88.6%, respectively, indicating that they strongly agree or agree. In contrast, only 44.4% of the Computer Science Subfield students selected agree with no students selecting strongly agree. Notably, 44.4% of the Computer Science Subfield students selected strongly disagree or disagree.

It is interesting that the oral presentation proficiencies do not mirror the technical documentation results. While BS and BA Computer Science students oral presentation results align with their technical documentation proficiency, with 93.1% and 91.4%, respectively, indicating that they strongly agree or agree, a higher percentage of Computer Science Subfield students selected strongly agree or agree for oral presentations compared to technical documentation, at 77.7% (See Figure 3).

In relation to the employer requirements analysis, it is notable that oral communication skills were mentioned in the highest percentage of job ads (19.5%), which aligns with the relatively high proportion of computer science subfield students—77.7%—who reported strong agreement or agreement in their oral presentation proficiency.

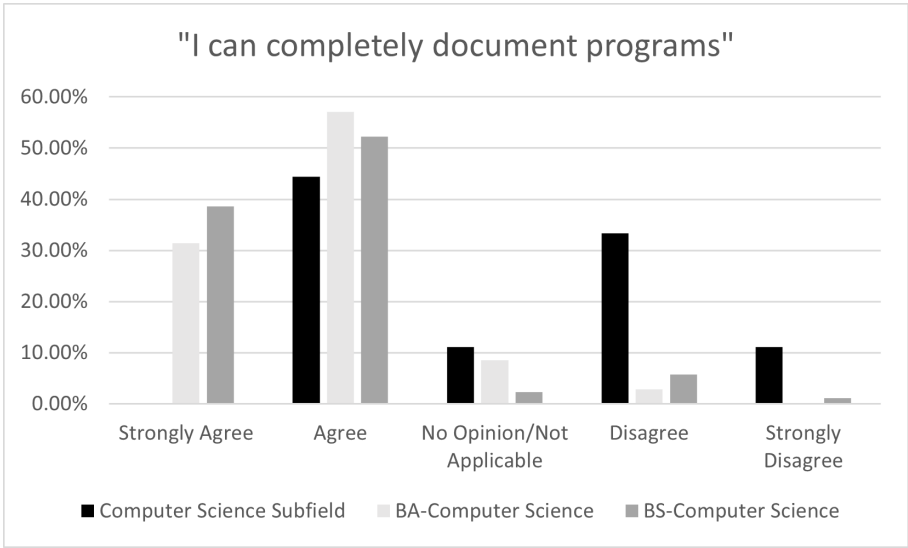


Figure 2: Frequency Distribution: Technical Documentation

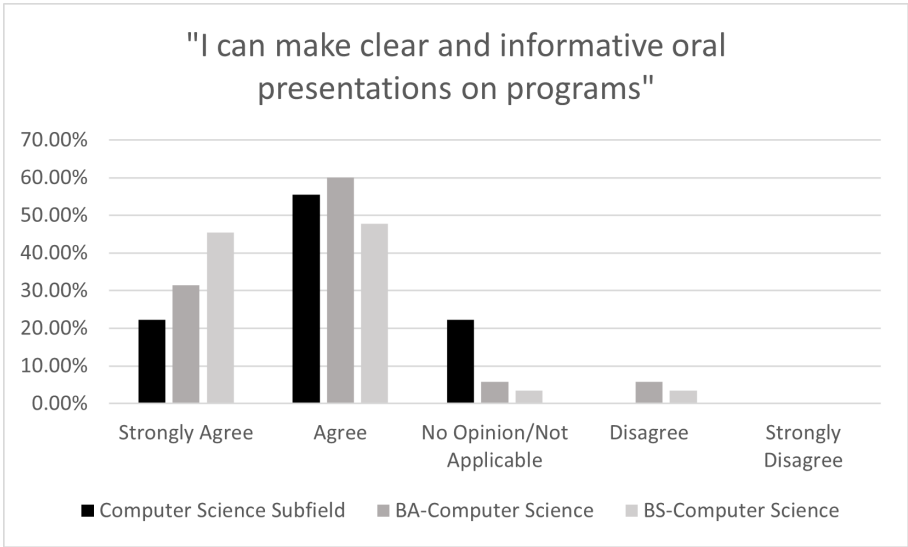


Figure 3: Frequency Distribution: Oral Presentation

5 Conclusion

In conclusion, this study highlights important differences between employer expectations and undergraduate skill development across computer science degree pathways. While traditional computer science programs continue to empha-

size software engineering competencies, job postings for specialized sub-majors place comparatively greater value on oral communication skills. These findings suggest that institutions should more carefully align curricula across all computer science degrees to reflect the evolving demands of the technical job market. Ensuring that graduates—regardless of degree option—develop a balanced combination of software development and communication skills is essential for improving workforce readiness and employer satisfaction.

References

- [1] Rea Lavi, Marina Tal, and Yehudit Judy Dori. “Perceptions of STEM alumni and students on developing 21st century skills through methods of teaching and learning”. In: *Studies in Educational Evaluation* 70 (2021), p. 101002.
- [2] James Foley and Thomas Standish. “Report of the NSF undergraduate computer science education workshop”. In: *ACM SIGCSE Bulletin* 20.3 (1988), pp. 57–64.
- [3] Albert Bandura. *Self-efficacy: The exercise of control*. New York: W. H. Freeman, 1997.
- [4] ABET Computing Accreditation Commission. *Criteria for Accrediting Computing Programs*. <https://www.abet.org>. Accessed: 2026-01-23. 2023.
- [5] Ahmad Z. Abualkishik et al. “Outcomes-Based Assessment and Lessons Learned in ABET-CAC Accreditation: A Case Study of the American University in the Emirates”. In: *Mobile Information Systems* 2022.1 (2022), p. 1595126. DOI: 10.1155/2022/1595126.
- [6] National Association of Colleges and Employers. *Career Readiness Competencies*. <https://www.nacweb.org>. Accessed: 2026-01-23. 2023.
- [7] Christopher DuPre and Karen Williams. “Undergraduates’ Perceptions of Employer Expectations”. In: *Journal of Career and Technical Education* 26.1 (2011).
- [8] Rita Garcia et al. “An International Examination of Non-Technical Skills and Professional Dispositions in Computing – Identifying the Present Day Academia-Industry Gap”. In: *2024 Working Group Reports on Innovation and Technology in Computer Science Education*. ITiCSE 2024. Milan, Italy: Association for Computing Machinery, 2025, pp. 124–174. ISBN: 9798400712081. DOI: 10.1145/3689187.3709610. URL: <https://doi.org/10.1145/3689187.3709610>.
- [9] Catherine Caballero and Arlene Walker. “Work readiness in graduate recruitment and selection: A review of current assessment methods”. In: *Journal of Teaching and Learning for Graduate Employability* 1 (Aug. 2010), pp. 13–25. DOI: 10.21153/jtlge2010vol1no1art546.
- [10] Gary McClelland and William Klemm. “Using indirect measures for outcomes assessment”. In: *Assessment Update* 30.5 (2018), pp. 8–11.
- [11] Karen Sparck Jones, Peter Willett, et al. *Readings in information retrieval*. Morgan Kaufmann, 1997.
- [12] Jaime Carbonell et al. “Context-Based Machine Translation”. In: *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*. Cambridge, Massachusetts, USA: Association for Machine

Translation in the Americas, Aug. 2006, pp. 19–28. URL: <https://aclanthology.org/2006.amta-papers.3>.

Reviewers — 2026 CCSC Midsouth Conference

Rickey Casey	<i>Campbellsville University</i>
Kriti Chauhan	<i>Austin Peay State University</i>
Anirudh Reddy Cingireddy	<i>East Central University</i>
Adrienne Decker	<i>University at Buffalo</i>
Indira Dutta	<i>St Cloud State University</i>
Tolga Ensari	<i>Arkansas Tech University</i>
Gabriel Ferrer	<i>Hendrix College</i>
Robin Ghosh	<i>Arkansas Tech University</i>
Olivera Grujic	<i>California State University Stanislaus</i>
Mario A.M. Guimaraes	<i>Southeast Missouri State University</i>
Susan Hammond	<i>Lipscomb University</i>
Brian K Hare	<i>University of Missouri – Kansas City</i>
Mark Holliday	<i>Western Carolina University</i>
Charles Glen Hoot	<i>Northwest Missouri State University</i>
David Hovemeyer	<i>Johns Hopkins University</i>
Charles Donald Lake	<i>Coastal Alabama Community College</i>
Rao Li	<i>University of South Carolina Aiken</i>
Edward Lindoo	<i>Regis University</i>
Sean McCulloch	<i>Ohio Wesleyan University</i>
David Middleton	<i>Arkansas Tech University</i>
Muhammad Rahman	<i>Clayton State University</i>
Nilanjana Raychawdhary	<i>Auburn University</i>
Kenneth Rouse	<i>LeTourneau University</i>
Abdus Salam Siddique	<i>Arkansas Tech University</i>
Scott Sigman	<i>Drury University</i>
Takako Soma	<i>Illinois Collage</i>
Dwayne Towell	<i>Lipscomb University</i>
Karen Works	<i>Florida State University Panama City</i>
Yingbing Yu	<i>Austin Peay State University</i>
Mian Afzaal Zahoor	<i>Saint Louis University</i>