The Journal of Computing Sciences in Colleges

Papers of the 39th Annual CCSC Eastern Conference

October 20th-21st, 2023
Bay Atlantic University
Washington, DC

Baochuan Lu, Editor
Southwest Baptist University

George Dimitoglou, Regional Editor
Hood College

Volume 39, Number 3

October 2023

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

Table of Contents

The Consortium for Computing Sciences in Colleges Board of Directors

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

Scott Sigman, President (2024), ssigman@drury.edu, Mathematics and Computer Science Department, Drury University, Springfield, MO 65802.

Bryan Dixon, Vice President/President-Elect (2024), bcdixon@csuchico.edu, Computer Science Department, California State University Chico, Chico, CA 95929.

Baochuan Lu, Publications Chair (2024), blu@sbuniv.edu, Division of Computing & Mathematics, Southwest Baptist University, Bolivar, MO 65613.

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

Cathy Bareiss, Membership Secretary (2025), cathy.bareiss@betheluniversity.edu, Department of Mathematical & Engineering Sciences, Bethel University, Mishawaka, IN 46545.

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.

David R. Naugler, Midsouth Representative (2025), dnaugler@semo.edu, Brownsburg, IN 46112.

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

Mark Bailey, Northeastern Representative (2025), mbailey@hamilton.edu, Computer Science Department, Hamilton College, Clinton, NY 13323.

Shereen Khoja, Northwestern Representative(2024), shereen@pacificu.edu, Computer Science, Pacific University, Forest Grove, OR 97116.

Mohamed Lotfy, Rocky Mountain Representative (2025), mohamedl@uvu.edu, Information Systems & Technology Department, College of Engineering & Technology, Utah Valley University, Orem, UT 84058.

Tina Johnson, South Central Representative (2024), tina.johnson@mwsu.edu, Department of Computer Science, Midwestern State University, Wichita Falls, TX 76308.

Kevin Treu, Southeastern Representative (2024), kevin.treu@furman.edu, Department of Computer Science, Furman University, Greenville, SC 29613.

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

CCSC National Partners

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

Gold Level Partner

Rephactor
ACM2Y
ACM CCECC

Welcome to the 2023 CCSC Eastern Conference

Welcome to the Consortium for Computing Sciences in Colleges (CCSC) Eastern's 39th Annual Regional Conference, hosted in the nation's capital this year by Bay Atlantic University, Washington, DC.

On behalf of the CCSC Eastern Region and the BAU conference committees, we greatly appreciate all the submissions for papers, posters, workshops, tutorials, and nifty ideas. All committees for this year's conference began to work with considerable effort at the beginning of 2023 to ensure the conference's success and a warm welcome to all parties.

While the CCSC committee intensively assessed the materials to ensure the highest quality of the materials selected for presentations and demonstrations, the BAU team, faculty, staff, and students worked tirelessly to prepare the locations for the event to the best of their ability. Accordingly, the two-day event offers carefully planned sessions to ensure compatibility of topics and easy access throughout the campus.

It is also a great honor to have Captain Warren D. Judge, Commanding Officer of Coast Guard Training Center Cape May, New Jersey, who also served as the Engineering Services Division Chief in Portsmouth, Virginia, for the Command, Control, Communications, Computers, Cyber, and Intelligence Service Center, as the conference's keynote speaker, and Mr. Richard Warren, a Business Information Systems and Technology Executive who has worked in the industry for more than 20+ years, as our Banquet speaker.

The committee is confident that audiences will gain tremendous knowledge while enjoying the presentations both of our honored speakers will bring. Besides, the topics of presentations carefully reviewed and selected will provide a variety in the realm of computing technology and curricula that significantly relate to audiences' various interests.

This year, the conference's steering committee received tremendous interest in paper submission, and the committee members assembled in person at BAU to finalize the decision to ensure the quality of topics in each submission type and track. Accordingly, we hope that the 39th CCSC Eastern Regional Conference will benefit all participants in gaining knowledge and experience as it has always been in all previous years.

We received 48 faculty papers and 27 were accepted (56.25%) after double blind peer review and 13 faculty posters. Student submissions were at 6 papers and 10 posters. Along these numbers, we accepted 2 workshops and 7 Nifty Ideas/Lighting Talks.

Lastly, as the chair of this year's conference, I am very grateful and honored to work alongside all committee members and the BAU team to accomplish all tasks and preparations that aim to ensure this prestigious regional forum for computing sciences offers a beneficial and amazing journey to all.

We look forward to meeting you in the nation's capital at Bay Atlantic University and greatly appreciate your participation and attendance.

Pipop Nuangpookka
Bay Atlantic University
Conference Chair

2023 CCSC Eastern Conference Committee

Pipop Nuangpookka, Chair	Bay Atlantic University
George Dimitoglou, Papers, Regional Editor	Hood College
John Wright, Papers	Juniata College
Natalia Bell, Posters	Marymount University
Susan Conrad, Posters	Marymount University
Karen Anewalt, Posters	University of Mary Washington
Ian Finlayson, Posters	University of Mary Washington
TJ Highley, Panels, Workshops, and Tutorials/Nifty Ideas	La Salle University
Steven Kennedy, Programming Contest	Frostburg State University
David Hovemeyer, Programming Contest	Johns Hopkins University
TJ Highley, Programming Contest	La Salle University
Pranshu Gupta, ConfTool Registration and Submission System	DeSales University
Michael Flinn, Regional Board Representative	Frostburg State University
Nathan Green, Regional Treasurer	Marymount University
Melissa Stange, 2024 Host,	Laurel Ridge Community College
John Wright, Web Site	Juniata College

2023 CCSC Eastern Conference Steering Committee

Nathan Green, Chair, Registration	Marymount University
Elizabeth Adams	James Madison University
Steven Andrianoff	St. Bonaventure University
Karen Anewalt	University of Mary Washington
George Benjamin	Muhlenberg College
Elizabeth Chang	Hood College
Vincent Cicirello	Stockton University
Susan Conrad	Marymount University
George Dimitoglou	Hood College
Michael Flinn, Regional Representative	Frostburg State University
Sister Jane Fritz	St. Joseph's College
Nathan Green, Interim Regional Treasurer (2019-21)	Marymount University
Pranshu Gupta	DeSales University
David Hovemeyer	Johns Hopkins University
John Meinke	University of Maryland University College
Karen Paullet	Robert Morris University
Donna Schaeffer	Marymount University
Jennifer Polack-Wahl	University of Mary Washington
John Wright	Juniata College

Current Challenges in Computing Education[*]

Panel Discussion

Susan Conrad
Marymount University, Arlington, VA 22207
sconrad@marymount.edu
George Dimitoglou
Hood College, Frederick, MD 21701
dimitoglou@hood.edu
Michael B. Flinn[†]and Jacob Morgan
Frostburg State University, Frostburg, MD 21532
{MFlinn, jlmorgan1}@frostburg.edu
Pranshu Gupta
DeSales University, Center Valley, PA 18034
Pranshu.Gupta@desales.edu
Zelalem Mengistu
Bay Atlantic University, Washington, DC 20005
zmengistu@bau.edu

Summary

Discussion about topics related to current issues in computing science education focusing on three themes: “That AI thing...”, “Post-Pandemic Strategies,” and “Partnerships.”

The first theme attempts to address the benefits, challenges, and practical applications of integrating Generative AI technologies, such as ChatGPT, Bard, and CoPilot, into educational settings. Exploration of academic honesty and intellectual property and strategies for how these AI tools can be utilized in classrooms, labs, student projects, assignments, academic programs, and even preparing students for future job opportunities.

[*] Copyright is held by the authors/owners.

[†] Panel Moderator

The second theme revolves around post-pandemic approaches and initiatives to explore aimed at re-engaging students in both classroom activities and extracurricular pursuits. Exploration of strategies to enhance undergraduate and graduate student participation in internships, research opportunities, and the unique challenges and characteristics of job hunting in the current educational and economic landscape.

The third theme highlights the significance of forging partnerships between educational institutions and industry stakeholders. Exploring campus ideas and efforts to establish and strengthen relationships with industry partners. Discussion on collaborative projects, research initiatives, mentorship programs, and ways to bridge the gap between academia and industry to benefit both students and the workforce.

The final theme is open-ended, encouraging attendees to contemplate additional questions that may initiate reflection on emerging trends, pedagogical challenges, technological advancements, and any other critical issues that computing science educators should address to stay effective and responsive in their roles.

Pushing the Four Envelopes[*]

Banquet Speech

Richard L. Warren
U.S. Department of Health and Human Services

Bio of the Banquet Speaker - Mr. Richard L. Warren

Richard Warren was born and raised in Michigan. In 1970, following his freshman year at Michigan State University, he made the pivotal decision to join the U.S. Navy. It was a choice that would define his life's trajectory. Richard quickly discovered his affinity for military service and embarked on a remarkable journey within the ranks. Over the years, he rose from Seaman Recruit to the esteemed rank of Chief Photographer's Mate. Notably, his dedication and talent led to a unique commission "from the ranks." His illustrious naval career came to a close in January 1992, culminating with his retirement as a Lieutenant.

Throughout his military tenure, Richard embarked on diverse assignments that spanned the breadth of the Navy's capabilities. He served in an oceanographic survey unit, sailed aboard an aircraft carrier, and commanded an amphibious command and control ship. Richard's contributions extended beyond the frontlines as he assumed roles as an acquisition program manager and, ultimately, the Technical Services Director at the Atlantic Intelligence Command. An exceptional facet of his military career was his creative prowess in multimedia production, earning him two gold and one silver medals from the International Film & Television Festival of New York.

Richard's quest for knowledge and personal growth did not wane in retirement. He diligently pursued his Bachelor's Degree, completing it through the University of the State of New York's Regents College program. This academic endeavor segued into a Master of Science in Management program with Troy State University. Although his career temporarily pulled him away from campus life, Richard's thirst for education persisted, leading him to commence his MBA studies in e-Business with the University of Phoenix in the spring of 2004. He successfully completed his MBA in e-Business in December 2005.

[*] Copyright is held by the author/owner.

Transitioning into a fulfilling post-Navy "second" career, Richard occupied various significant positions. He assumed the mantle of Vice President for Information Technology at a prominent magazine and catalog printing company. Moreover, he founded an Internet start-up within his corporate IT department, showcasing his entrepreneurial spirit. His professional journey featured multiple executive roles, such as Vice President, Senior Vice President, and Chief Officer, in both small and large IT consulting firms. Richard's stature in the industry was underscored by his membership on five of Microsoft's global Partner Advisory Councils, including the prestigious E-Business Partner Advisory Council.

Richard's client portfolio during his second career was illustrious, encompassing organizations like Martha Stewart Living Omnimedia, Ford Motor Company, the Bank of New York, Pfizer, the U.S. Department of Agriculture, Electronics Boutique, The Wedding Channel, U.S. Air Force E-Procurement initiatives, and Sky Publishing Company. His expertise extended far beyond technical matters, as evidenced by his role in launching Martha Stewart's original website and providing exclusive sitebuilding services.

In September 2006, Richard embarked on a new executive role as Chief Technology Officer for an online marketing and lead management provider in Virginia Beach, Virginia. He successfully oversaw a technology migration to a highly virtualized hosting environment, upgraded the corporate infrastructure, and instilled a disciplined Application Lifecycle Management approach using agile processes (SCRUM) before moving on to new horizons.

In 2009, a third phase of Richard's career began when he accepted a position at the Environmental Protection Agency (EPA). He actively pursued an Enterprise Project Management agenda within the agency's Project Management Office while concurrently pursuing a Master of Science in Information Technology with a specialization in Project Management. Currently, Richard is in pursuit of a Ph.D. in Management at Walden University.

In late 2013, Richard transitioned to a senior advisory role and E-Business Strategist at the Program Support Center within the U.S. Department of Health and Human Services. His initial focus was resuscitating a faltering e-business transformation initiative, and he successfully launched the revamped e-business platform. Presently, his endeavors revolve around aiding PSC in deploying a Project Management Office, implementing an organizational SharePoint Online solution, and establishing an integrated project management information system using Microsoft Project Online.

Beyond his professional roles, Richard's impact extends to corporate governance. He has served on the Board of Directors for several companies, providing invaluable independent oversight and strategic guidance in the realm of e-business initiatives. Richard Warren's remarkable journey exemplifies a lifetime of dedication, innovation, and unwavering commitment to excellence.

GitKit: Teaching Git and GitHub/GitLab Workflow in an Authentic Context[*]

Workshop

Grant Braught[1], Stoney Jackson[2]
[1]Dickinson College
Carlisle, PA 17013
braught@dickinson.edu
[2]Western New England University
Springfield, MA 01119
hjackson@wne.edu

The GitKit facilitates teaching Git and GitHub workflow in the context of an authentic Free and Open Source Software (FOSS) project. It is appropriate for use in software development courses ranging from high school through college. The GitKit is a snapshot of a FOSS project's artifacts (codebase(s), issues, etc.) packaged with student learning activities, an instructor guide, and a containerized development environment. The GitKit can be used to provide a light introduction in a few class sessions, or a more comprehensive experience over 4-6 sessions. Participants will gain hands-on experience with the GitKit from both the student and instructor perspectives.

The Blockchain Art Simulation (BARTS) and Experiential Exercises[*]

Workshop

Sean Sanders and George Sanders
SUNY-Buffalo
Buffalo, NY 14260
{spsander, mgtsand}@buffalo.edu

This workshop introduces the online Blockchain ART Simulation (BARTS). The simulation illustrates the interrelationship of blockchain mining, cryptocurrency, non-fungible tokens (NFT), and hashing concepts. We developed the simulation to gamify the learning process where students participate in a simulation for buying and selling NFT drawings. Experiential online exercises are also presented, illustrating hashing concepts for validating blockchain transactions. The workshop participants engage in the mining process using their phones. We have used this material successfully with over 500 high school students to graduate students.

Developing Identity-Focused Program-Level Learning Outcomes for Liberal Arts Computing Programs*

Tutorial

Jakob Barnard[1], Grant Braught[2], Janet Davis[3],
Amanda Holland-Minkley[4], David Reed[5], Karl Schmitt[6],
Andrea Tartaro[7], James Teresco[8]
[1]University of Jamestown, Jakob.Barnard@uj.edu
[2]Dickinson College, braught@dickinson.edu
[3]Whitman College, davisj@whitman.edu
[4]Washington & Jefferson College, ahollandminkley@washjeff.edu
[5]Creighton University, DaveReed@creighton.edu
[6] Trinity Christian College, Karl.Schmitt@trnty.edu
[7]Furman University, andrea.tartaro@furman.edu
[8]Siena College, jteresco@siena.edu

The SIGCSE Committee on Computing Education in Liberal Arts Colleges (SIGCSE-LAC Committee) has found that liberal arts and small colleges approach design of their computing curricula in unique ways that are driven by institutional mission or departmental identity. This impacts how faculty at these colleges adopt curricular guidelines such as the current ACM/IEEE-CS CS2013. The committee is developing guidance, informed by its sessions at recent CCSC and SIGCSE conferences, to help with the design and/or revision of CS curricula in liberal arts contexts. This will ultimately be included in the committee's article in the Curricular Practices Volume that will be released as a companion to the new ACM/IEEE-CS/AAAI Computer Science Curricula guidelines (CS2023). Curricular guidelines like CS2013 or CS2023 inform curriculum design but are balanced with the vision for a program, departmental strengths, locale, student populations and unique academic experiences. The desire to craft distinctive curricula, combined with the size of prior curricular recommendations, requires an assessment of trade-offs between achieving full coverage of curricular recommendations and a school's other priorities.

SIGCSE-LAC's guidance will encourage faculty to reflect on their programs and the role of CS2023, beginning with their institutional and departmental priorities, opportunities and constraints.

The specific goal of this session is to help participants develop program-level learning outcomes that align with the unique features of their programs. Following an overview and brief discussion of the newest CS2023 draft, participants will begin working through a preliminary version of the committee's reflective assessment process. This process is framed by a series of scaffolding questions that begin from institutional and departmental missions, identities, contexts, priorities, initiatives, opportunities, and constraints. From there, participants will be led to identify design principles for guiding their curricular choices including the CS2023 recommendations. Examples gathered from the committee's previous CCSC and SIGCSE sessions will be available to help to articulate identity and program design principles, which will then be used for the identification of identity-focused program-level learning outcomes. Participants will leave the session with a better understanding of how CS2023 can impact their programs and a jumpstart on the entire reflective assessment process. Feedback on the process and this session are welcome and will be used to refine the committee's guidance prior to its publication in the CS2023 Curricular Practices volume.

Binary Patterns*

Tutorial

Peter Henderson
Butler University
Indianapolis, IN 46208
phenders@butler.edu

Binary is used widely in computing, and therefore important for students to understand its basic concepts. It is used for the representation of information in modern computers, including, representing numbers and operations on numbers, both binary and floating point, encoding characters, digital image representation, error detection and correction, Boolean logic, machine code, information storage, UPC bar codes, etc.

In this tutorial the idea of binary patterns, as an overarching approach to understanding binary concepts are introduced. Morse code, train whistles, and braille are examples of binary patterns which can be used to explain and motivate binary concepts. Character representations follow patterns, for example, in ASC-II digits 0-9 are represented sequentially, as are upper and lower case letters. Basic binary number representations follow well defined patterns, as do signed number representations such as 2's and 1's complement, used for computations in computer systems.

Techniques and various kinesthetic learning activities for student discovery learning of basic binary patterns will be introduced. These start with identifying patterns in sequences of 0's and 1's, leading to the discovery of patterns in binary number representations and binary number arithmetic. Binary error detection pattern discovery activities, including the parity card flipping challenge from CS Unplugged, are presented.

Finally, the use of error correction in data communication is introduced. This includes how CD's and DVD's work, and data transmission in noisy environments, such as space and satellite communications.

Performance Testing of a Web Application Using Azure Serverless Functions and Apache JMeter[*]

Tutorial

Hardeep Kaur Dhalla
University of Wisconsin–Stevens Point
Stevens Point, WI 54481-3897
hdhalla@uwsp.edu

Software testing is one of the most critical phases in the software development life cycle. It is of utmost importance to learn how to verify and validate a software application. Moreover, the performance of the software is also pivotal for the success of any software application. This tutorial aims to provide participants with a comprehensive understanding of performance testing methodologies using Apache JMeter, with a specific focus on leveraging Azure Serverless Functions as endpoints. I will demonstrate the use of Apache JMeter automation testing tool to create test plans and to generate artificial workload for real-world scenarios of performance testing. Moreover, the participants will learn the impact of resource allocation variations using cloud infrastructure on the performance of web applications.

[*] Copyright is held by the author/owner.

“I, ChatBot”: Co-Teaching Cybersecurity Courses With Generative AI[*]

Tutorial

Karla Carter
Bellevue University
Baltimore, MD 21215
kcarter@bellevue.edu

This tutorial is for computing science faculty who are intrigued by the notion that generative AI, such as OpenAI’s ChatGPT or Google’s Bard, can enhance the way we teach and students learn cybersecurity. Rather than questioning if faculty and students should use generative AI in the classroom, you’re asking how faculty and students can use generative AI appropriately and responsibly in the classroom. Our students deserve to understand the tools shaping their future; generative AI is not going away and we need to prepare our students for a future where not knowing how to write generative AI prompts isn’t an option.

Parlor tricks including having generative AI rewriting Sherlock Holmes’ “The Adventure of the Dancing Men” as a modern cybersecurity tale involving reverse engineering malware and casting Professor Moriarty as a mastermind black-hat hacker aside, generative AI can be a useful academic brainstorming companion for both faculty and students. Faculty can use it as an assistant to construct real-world cybersecurity scenarios that can engage students. Students can partner with generative AI to jumpstart projects and improve their communication skills – they may not know what sort of memo to send to the CISO, but generative AI likely does and can help take their thoughts and polish the prose to meet the requirements. This isn’t just about using generative AI - it’s about understanding it, too. We’ll consider the use of generative AI as a tool that complements and enriches - not replaces – teaching and learning abilities. The best-case scenario for any AI assistant is to free humans from tedium and leave more time for creative and critical thinking.

We'll begin the tutorial by going over the basics of generative AI and what its capabilities and drawbacks (e.g. hallucination) are. After that you'll get hands-on time to write prompts, both guided and independent. We'll explore using a series of prompts to devise assignments that serve a dual purpose: teaching the ins and outs of cybersecurity, and through purposeful transparency modeling the usage of and sparking an appreciation for appropriate and responsible generative AI usage. We will role play students completing these assignments, as well, in order to understand what sort of instructional support they will need in terms of digital literacy and fact-checking. Participants will leave with handouts that enable them to continue their explorations at home.

Tips and Tricks for Teaching Switch..Case Structure in Python*

Nifty Idea

Penn Wu
DeVry University
Sherman Oaks, CA 91403
pwu@devry.edu

Selection structure is an important topic of Python-based programming courses. In traditional programming languages like C++, Java, C#, and so on, "switch..case" structure is a form of selection structure used to obtain the value of a variable or evaluate an expression in order to determine the control flow of program execution. Interestingly, Python does not provide any official "switch..case" structure as of its latest version–Python 3.11. However, the Python version 3.10 introduces a new feature called "structural pattern matching" which is arguably the Python's implementation of the "switch..case" structure.

This presentation will discuss: (1) how Python's new "match..case" structure works, (2) how Python's "match..case" structure is compatible to the "switch..case" structure of C++, C# and Java, (3) how the "match..case" structure may go beyond the traditional "switch..case" statement; (4) how other Python structures can be used to simulate the "switch..case" structure, and (5) why the "match..case" structure is an efficient alternative for if-else statements. The presenter will also provide sample instructional materials with Python codes and hands-on learning activities to share tips and tricks for teaching "switch..case" structure in Python.

*Copyright is held by the author/owner.

Dear Neighbor: Blending Computer Science, Experiential Learning, And Community Outreach For Real-world Application[*]

Nifty Idea

Andrea Marie Wentzell
Chestnut Hill College
Philadelphia, PA 19118
wentzella@chc.edu

The “Dear Neighbor” is a crucial value and part of the mission for Chestnut Hill College. Starting in Spring 2021, the Computer Science Department developed an On-Site Internship Program that blends mission, experiential learning requirements, and community outreach to offer real-world experience in website and mobile application development. The program focuses on serving the “Dear Neighbor,” working with neighborhood organizations, often nonprofits, in Philadelphia. Many organizations (or clients) are committed to social change, peace, environmental impact, and service.

Over the last seven semesters, students have completed internships working one-on-one with clients while also being further mentored by professors. The program is designed for clients without the funds to employ someone or have their website/mobile application created by firms. While furthering their programming skills, students are simultaneously developing problem-solving, communication, and independent work skills. In addition, students often leave the experience with a launched website or mobile application, perfect portfolio collateral.

Student and client survey results will be shared, along with the program’s setup, development, and implementation.

Computer Science Capstone: How to Encapsulate Four Years into Four Credits[*]

Nifty Idea

Ruth Lamprecht and Scott Weiss
Mount St. Mary's University
Emmitsburg, MD 21727
{r.e.lamprecht, sweiss}@msmary.edu

We present our methodology for a four-credit course that serves as a capstone to the Computer Science B.S. degree program. Our course is split between two semesters as a 1-credit course in the fall and a 3-credit course in the spring. The general concept is that students use the fall to pick a project and develop a proposal that is then implemented during the spring. But there is much more that we include.

Theme: Each year we select a theme to help guide the students in selecting a project, and to give some commonality between a very diverse set of projects.

Ethics: In addition to the project, we include student-led discussions of ethics and how they pertain to technology and their future careers. A textbook is selected for reference, chosen to have a wide range of issues presented in an easily read manner. A small group of students are assigned to each chapter to present a short recap of the material and then to lead a class discussion.

Assessments: Another component of our course is several assessments used for the yearly department report. Students participate in a programming contest with problems written by the department faculty, complete the ETS Major Field Test for Computer Science, and write a 2-3 page reflection essay on their time at the university.

Presentations: Over the full academic year, students make several presentations, including the ethics discussions. Each student is required to find an article of interest that has been published in an ACM journal to engage students in current topics and practice reading and presenting articles of this level. Students make several presentations about their final project, including a culminating presentation at a university-wide student research conference.

Oral Exams in CS-Education: Pros and Cons in the Age of AI Assisted Programming*

Nifty Idea

Ed Novak[1] and Peter Ohmann[2]
[1]Franklin and Marshall College
Lancaster, PA 17603
[2]College of Saint Benedict & Saint John's University
St. Joseph, Minnesota 56374/Collegeville, Minnesota 56321
enovak@fandm.edu, POHMANN001@csbsju.edu

As AI coding tools proliferate through the computer science community, oral exams present a compelling way to assess the skills of students. Unfortunately, oral exams also present some difficulties, particularly for large class sizes.

Tools based on Large Language Models (LLMs), like ChatGPT and GitHub Copilot, show how programming can be done in part using natural language interactions.

With the growth of computing throughout society, this points to a future in which communication skills are increasingly important for computer science undergraduates. Oral exams are a natural way to build and assess students' communication skills. Further, AI-based tools challenge traditional assessment methods such as homework assignments and digital exams. When students can simply ask an AI tool to complete and/or debug their assignments, oral exams provide a way to ensure that students' responses are their own.

Although some research in CS oral exams exists, two problems that most educators perceive when considering oral exams are (1) planning the timing and logistics and (2) evaluating students responses in an ethical and unbiased way. Traditional exams can be done in parallel, but oral exams are typically structured as one-on-one conversations between the educator and each student. Such setups do not easily scale in the context of growing enrollments. Evaluating oral exams is also challenging due to the potential for educator bias, since oral exams generally cannot be done anonymously. Finally, it is not trivial to produce and enforce a rubric during a live oral exam conversation.

This lighting talk will present pros and cons of oral exams based on past research and experiences, and consider what research projects are necessary to make oral exams effective and attractive to CS educators. The intended audience is those teaching or administering computer science topics (especially programming) for students from high-school through college.

How to Integrate ChatGPT into the CS1/CS2 Sequence[*]

Nifty Idea

Junyi Tu
Salisbury University
Salisbury, MD 21801
jxtu@salisbury.edu

ChatGPT is one of most revolutionary technologies and fiercely debated on the benefit and disaster of its effect on human society. How to guide students to use this new technology is an inevitably topic as CS educators. This nifty idea includes ways to integrate ChatGPT into the CS1/CS2 sequence, how to guide students to use ChatGPT in a constructive way, instead of cheating on their homework.

Customized Taan Generation in Hindustani Classical Music: A Relational Approach*

Nifty Idea

Hardeep Kaur Dhalla
University of Wisconsin–Stevens Point
Stevens Point, WI 54481
hdhalla@uwsp.edu

Composing a taan (a short tune or pattern of notes) spontaneously while performing Hindustani classical style music is the traditional way which requires years of experience and practice. It can be a daunting and time-consuming task for a composer who is not classically trained, to come up with even a few taans that might beautify the composition. I am experimenting with a desktop application in Java that stores the attributes of a given raga or a composition in a relational model and exploits the rules and the logical structure of the Hindustani classical music's framework to generate millions of possible taans which can then be systematically filtered and narrowed down to the set of taans that satisfy user's desired criterion.

Student Perceptions and Attitudes of Generative AI in Higher Education and the Workplace[*]

Nifty Idea

Joan E. DeBello, Mehmet Soydan, John Mussalli,
Shivani Rambaran, Payal Moorti,
Christina Mattheopoulos, Sebastian Torres
St. John's University
Queens, NY 11439
debelloj@stjohns.edu, {mehmet.soydan22, john.mussalli22}@my.stjohns.edu
{shivanni.rambaran21, payal.moorti21}@my.stjohns.edu
{christiana.mattheopoulos22, sebastian.torres22}@my.stjohns.edu

This is a work in progress of a research project. We are working on a study of generative AI and its impact on higher education and the work place. In this session, we will discuss the ongoing efforts of the study and introduce the survey questions and literature review that will be used for the study. The goal is to determine student attitude, perceptions, and belieff of generative AI and its impact on their experiences in higher education and how it will impact their careers in the future.

A Practical Approach to Increasing Female Enrollment in Advanced Computer Science Courses[*]

Steven Schiff[1], Scott Frees[2]
[1]Computer Science
South Brunswick High School
Monmouth Junction, NJ 08852
steven.schiff@sbschools.org
[2]Department of Computer Sciesnce
Ramapo College of New Jersey
Mahwah, NJ 07430
sfrees@ramapo.edu

Abstract

Retention of female students in advanced computer science is a well recognized weakness in both K-12 and higher education. The absence of "a feeling of belonging" is often indicated as an important obstacle for students in underrepresented groups. To promote a sense of belonging, we created a "buddy system" to help encourage female students to continue on to the most advanced computer science courses in a high school curriculum. Female students that participated were able to select which students would be in their class. This initiative doubled the enrollment of female students.

1 Introduction

There has been a noticeable trend across all levels of computer science education where the percentage of females in computer science classes generally decreases as students move through the curriculum. We have observed this trend first hand at South Brunswick High School (SBHS), a public school located in Monmouth Junction, New Jersey. Approximately 2,785 students actively attend SBHS, which serves students from grades 9-12, and provides a free public education to the residents of South Brunswick Township. There are currently 1362 students that identify as female, 1420 that identify as male, and 3 that identify as non-binary. SBHS offers an innovative curriculum for advanced computer science courses, operating as the SBHS Computer Science Academy (CS Academy). The Academy is open to all students, and consists of multiple years of Computer Science coursework. Students who complete the Academy's curriculum can earn up to 12 college credits in Computer Science, through articulation programs with Ramapo College of New Jersey. At the onset of the Academy, the gender breakdown was 77% identifying as male, and 23% identifying as female. Approximately 200 freshman students enroll in the entry-level Computer Science course each year, and typically less than half will complete the entire sequence. Some of these students will elect to enroll in the academy.

Many of the female students in, or considering enrolling in, the academy have voiced the concern that they do not want to be the only girl in the class. This generally becomes a prominent concern as the student moves further into the program. Earlier courses in the sequence are much more balanced when examined via a gender perspective.

In a 2014 town hall Q and A session, Mark Zuckerberg (founder of Facebook/Meta) addressed females in computer science directly when he said, "I heard one person put it this way, that the reason why girls don't go into computer science is because there are no girls in computer science" [7]. To test this hypothesis, and attempt to create a more even gender balance in our Academy, we piloted a "buddy system" program, where female students could register for classes together. This system was used exclusively for our Computer Science Academy students, and resulted in an increase of 108% in female enrollment in the academy. This created a much more balanced second year of the program where 40% of the students are female and 60% are male.

2 Background

Höhne and Zander spoke about a concept of belonging uncertainty in regards to gender and computer science. They state, "A well-established predictor of minority students' academic underachievement is the worry to not "fit into" the respective academic environment: belonging uncertainty." [2] and it is this "belonging uncertainty" that may help to be addressed with a buddy system.

Continuing with this same idea of belonging, Master, Cheryan, and Meltzoff state, “However, our findings demonstrate that cues indicating that they are welcome and belong in this environment can increase girls’ self-reported interest in computer science, despite these prevailing stereotypes. The current studies showed that redesigning the classroom signaled a different image of computer science that encouraged girls to enroll in these important classes. Intentionally designing and changing high school physical environments (classrooms, computer labs, and offices) may play a significant role in communicating a feeling of belonging to girls and help to reduce current gender disparities in STEM courses.” [1]. Although not a physical environment, the gender makeup of the class is certainly a characteristic that students would likely take into account when registering.

Although there is a representation of women in some STEM jobs, there is still an under representation in computer science degree attainment. This can be seen from Fry, Kennedy, and Funk when they state, “Women earned 53% of STEM college degrees in 2018, smaller than their 58% share of all college degrees. The gender dynamics in STEM degree attainment mirror many of those seen across STEM job clusters. For instance, women earned 85% of the bachelor’s degrees in health-related fields, but just 22% in engineering and 19% in computer science as of 2018” [5]. Unsurprisingly this also carries on into the workforce representation. They also shared that, “Women account for 25% of those working in computer occupations. The share of women in this fast-growing occupation cluster declined from 2000 to 2016 and has remained stable since then” [5].

3 Course Sequence

Shown in Figure 1 are two sample course paths that a student might take at SBHS. There are a number of variations, this is simply a sample of what a pathway might look like for a student. Regardless of path, AP Computer Science is a prerequisite for Mobile Application Development, VR & Game Design, or Data Structures. For example, a student may skip a class junior year and still be allowed to take Data Structures senior year.

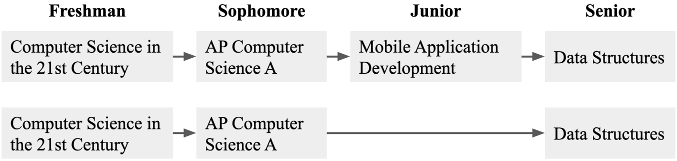

Freshman
Sophomore
Junior
Senior
Computer Science in the 21st Century
AP Computer Science A
Mobile Application Development
Data Structures
Computer Science in the 21st Century
AP Computer Science A
Data Structures

Figure 1: Sample course sequences for a CS student at South Brunswick High School.

In New Jersey there are no computer science requirements for graduation [4], and computer science classes are not required by the school district. This means that all of the students that enroll in our CS courses are doing this because they would like to take the course, not because it is required.

The only requirement to take Computer Science in the 21 st Century at SBHS is Algebra I. Any student that receives an 85% or higher in Algebra I would be eligible to take Computer Science in the 21 st Century. It should be noted that there is a non-21st Century version of the class that can be taken after freshman year, but that course has much lower enrollment, and if students start later in their high school career there is less time to take the advanced courses. Generally, the most interested students will take a computer science course freshman year, so the scope of this paper will be restricted to those students.

There are three courses open to general registration for students after completing AP Computer Science A (APCSA) at SBHS. There is a general recommendation for the sequence, but a student is allowed to skip a course if they are entering senior year, or if they have a strong preference.

SBHS offers a Computer Science Academy for students that are interested in computer science. Students that complete APCSA with an 85% or higher by the end of sophomore year are eligible to enter the academy. Students in the academy will take two of the most advanced computer science courses junior year and two more senior year. All of these courses will be completed as a cohort. This means that the same students will be in each of the advanced classes for academy students. This buddy system was introduced and used for the students entering the academy in the 2023-2024 school year.

Since the CS Academy requires a student to take all of the advanced courses for which APCSA is a requirement, this course acted as a foundational class, and is where we decided to enact change. There are a variety of other opportunities that the students in the academy can take advantage of as well. For example, students have the opportunity to participate in internships, attend CS related trips, take part in guest speaker lectures from industry specialists as well as SBHS alumni, and participate in competitions like Code Quest and CYBERQUEST® [3]. All students in the academy will take a capstone class, where they will create a culminating project that exhibits their mastery in a student selected area of computer science.

Students that apply are asked to write an essay about how they will work well in a community of CS students, and they make a two year commitment to the program when they sign up at the end of their sophomore year. A full listing of current requirements can be found on the Academy website for [6]. One of the most important aspects to keep in mind is that all of the students in the academy will take the four most advanced computer science courses

(Mobile Application Development, VR & Game Design, Data Structures, and the Capstone). Each cohort of students will take their CS courses together, and the sequence can be seen in Figure 2 [6]. The 2022-2023 school year was the first year that we ran the academy, and unsurprisingly the enrollment numbers were similar to the previously observed trends with female enrollment.

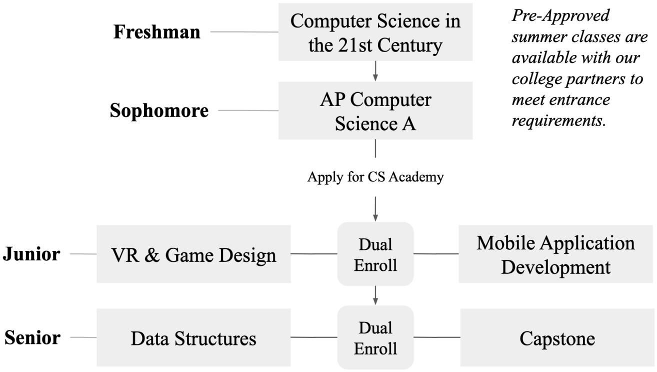
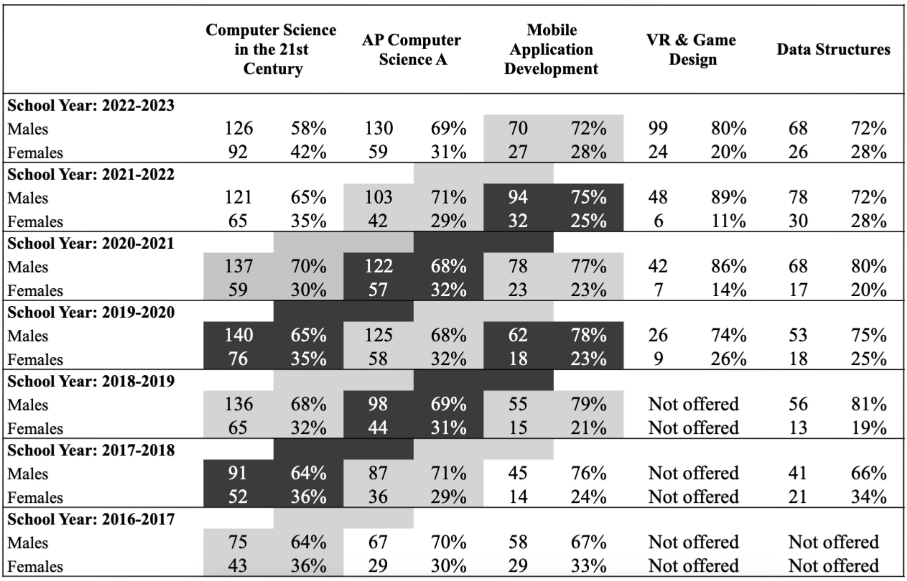

Freshman
Computer Science in the 21st Century
Pre-Approved summer classes are available with our college partners to meet entrance requirements.
Sophomore
AP Computer Science A
Apply for CS Academy
Junior
VR & Game Design
Dual Enroll
Mobile Application Development
Senior
Data Structures
Dual Enroll
Capstone

Figure 2: Computer Science Academy Course Sequence at South Brunswick High School.

4 Student Demographics

Student enrollment data by gender from 2016 to 2022 are provided in Figure 3. There is a general trend that can be seen from the data showing a lower retention rate for female students that progress through the courses. Generally, the number of students will decrease from one year to the next since not every student will elect to continue. This issue is more pronounced in female students, and seems to become more evident in the most advanced courses. These three courses (Mobile Application Development, VR & Game Design, and Data Structures) are all required for students in the Computer Science Academy. There is also the rare possibility that a student will place in from a summer class, transfer from another school, or move in from another class at the high school not listed above. This is likely why there are more male students enrolled in APCSA in 2018-2019 when compared to the perquisite course from the previous school year.

Although not guaranteed, most of the students will start with Computer Science in the 21 st Century and then progress to APCSA. From there it is possible that a CS student could progress to another one of the advanced courses, but most elect to take Mobile Application Development. Since this paper focuses on female students progressing into the most advanced courses offered, it should be clear from the Figure 4 that in the courses that follow APCS, there is a decrease in both the number of female students and the ratio of female to males. The specific course will make a difference in the percentage decline, but it seemed reasonable to select the most common course that students elect to take after APCSA.



	Computer Science in the 21st Century		AP Computer Science A		Mobile Application Development		VR & Game Design		Data Structures	
School Year: 2022-2023										
Males	126	58%	130	69%	70	72%	99	80%	68	72%
Females	92	42%	59	31%	27	28%	24	20%	26	28%
School Year: 2021-2022										
Males	121	65%	103	71%	94	75%	48	89%	78	72%
Females	65	35%	42	29%	32	25%	6	11%	30	28%
School Year: 2020-2021										
Males	137	70%	122	68%	78	77%	42	86%	68	80%
Females	59	30%	57	32%	23	23%	7	14%	17	20%
School Year: 2019-2020										
Males	140	65%	125	68%	62	78%	26	74%	53	75%
Females	76	35%	58	32%	18	23%	9	26%	18	25%
School Year: 2018-2019										
Males	136	68%	98	69%	55	79%	Not offered		56	81%
Females	65	32%	44	31%	15	21%	Not offered		13	19%
School Year: 2017-2018										
Males	91	64%	87	71%	45	76%	Not offered		41	66%
Females	52	36%	36	29%	14	24%	Not offered		21	34%
School Year: 2016-2017										
Males	75	64%	67	70%	58	67%	Not offered		Not offered	
Females	43	36%	29	30%	29	33%	Not offered		Not offered	

Figure 3: Computer Science Academy Course Sequence at South Brunswick High School.

5 The Buddy System

The buddy system allows a student to pick a friend or group of friends that will be in their next class or classes. We did not put a restriction on the number of students in a group, but we told the students to keep the size reasonable. A group of 20 students for a class of 25-30 students would likely create some gender imbalances when placing the groups into classes, but groups of 6 or 7 would work fine. Depending on the total number of enrolled students the groups can be placed into the different sections. This type of large group format also provides a level of confidence for the students. If one of students in the group decides to leave, most of the group will still remain.

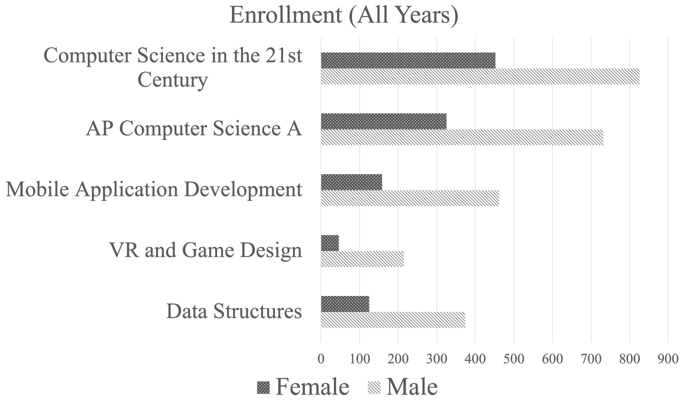


Figure 4: Total enrollment for all years, by gender in select CS courses at South Brunswick High School.

We reached out to all of the female students in APCSA via email and let them know that there would be a buddy system where they could select a friend or group of friends that they could take their course with if they enrolled in the CS Academy. We invited all of them to come to an informational session to hear about how the buddy system works, and asked their teachers to remind them to attend.

We invited the female students in the academy and some of the female students from independent courses to the information session. The conversation was facilitated by the Computer Science Department Chair, and the students spoke about their experiences in some of the advanced courses, and if applicable, the CS Academy.

The promise that other female students would join was not one that we could keep. We could hope that this would encourage female students to join, but in actuality, we could not make a guarantee. This was a problem since we wanted to create that certainty for the students. To create that guarantee, we added two additional elements to the meeting. First, we invited the staff member to participate in the meeting that creates the master schedule for school. He told the students that if they participated, he would create their schedules before anyone else's in the building. This would ensure that they are placed together, and there would not be any conflicts since their schedules were locked in first. Most students do not meet the individual that builds the master schedule for the school, and the hope was that this would build some confidence in our ability to deliver on our offer. The second element that we added was sharing a Google sheet with all of the female students that allowed them to create their groups. All of the students were able to view the document in real time and edit the document as they saw fit. The only information they needed to provide were their names, student ID, and a group number. All students in the same group were guaranteed to be in the same class. Unless a student decided to rescind admission to the academy, the students would have a reasonable guarantee of who would be in their groups, and the number of female students in the academy.

The practice at SBHS for course selection is that teachers will provide recommendations for the next most appropriate course. In addition to this, we made sure that announcements were made in all APCSA classes since this is the class that is the prerequisite to enter the academy. It is very unlikely that any female student in APCSA would have been unaware of the CS Academy in the initial year before the cohort system was introduced.

6 Who and Why

We decided to roll out the buddy system initiative during the 2023-2024 school year. Enrollment in the academy for the 2022-2023 school year consisted of 40 males and 12 females. The students enrolled in the academy are generally the students that would like to take all of our advanced courses. With the data trending towards lower enrollment for females in the later courses this seemed like a good place to try and create an initiative to increase female enrollment.

All of the classes in the academy trend lower with female enrollment, and all of the students that signed up would enroll in Mobile Application Development, VR & and Game Design, Data Structures, and the Capstone class. It is certainly possible that other sections of those courses have different breakdowns of male and female students, but we can monitor the sections that academy students are enrolled in.

The decision to initially roll this out to the female CS students was based on a few factors. As a pilot, we wanted to see if the buddy system would increase female enrollment in our highest level CS courses, but we believed that this would be based on the environment changing for female students. If we rolled this out to all students as a starting point, we were concerned that the female students would still feel they would be outnumbered, and this would nullify the idea that their environment would change. Once we determined if they buddy system worked, we would be able to examine the best way to roll it out as we move forward.

7 Results

A total of 25 female students signed up for the academy. 23 of the students signed up on the Google Sheet, and they broke themselves into 4 different groups. They created groups of 6, 7, and two groups of 5. Two of the students did not fill out the spreadsheet, so they will each be placed in groups by SBHS faculty. Equal grouping was not a requirement, but it does appear to have worked out well, and scheduling for equal groups should be relatively easy. This represented 40% of academy, and is noteworthy for several reasons. This was the first time that the ratio of female students to male students increased from the previous course, and there was a high conversion rate from APCSA. This is our target class where we look to draw students to enroll in the academy. It is possible that a student may elect to skip this class by self-studying or taking a summer class, but generally, this is where we advertise. The rate that females joined the academy increased from 29% to 42%.

As shown in Figure 5, the first year of the academy only had 12 female students, and the increase to 25 represents a 108% increase in the number of female students, and these students are going to take all of the advanced courses (Mobile Application Development, VR & Game Design, Data Structures, and the Capstone course). It should be noted that the VR course typically has low enrollment with female students, and all of the students in the academy will take this course. Just from enrollment numbers in the academy we can predict that we will have more female students in that class than any other year.

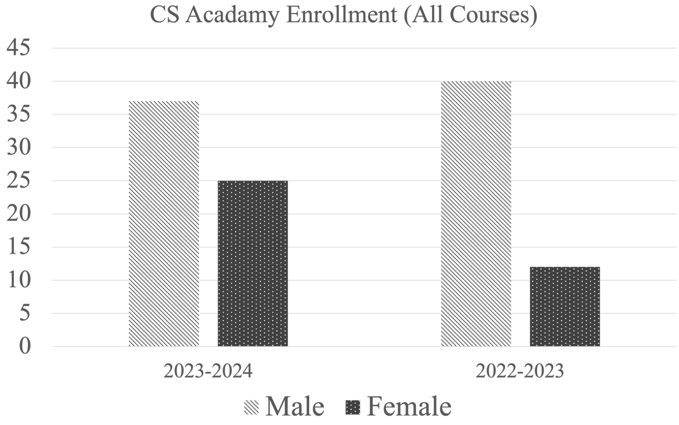


Figure 5: Academy enrollment at South Brunswick High School AY 2022 and 2023.

The buddy system has increased female enrollment in highest level courses, and has increased the rate that the females decide to enroll in the highest level CS classes.

8 Conclusion

The idea of a buddy system worked extremely well. Previously, there was never an increase in the percentage of female students when moving from APCSA to any of our other advanced courses, and this initiative will benefit not only a single course, but all of the highest level CS courses. There has also been some discussion at SBHS to see if this buddy system can be leveraged to help more groups of underrepresented students. This is something that the school will likely explore in more detail as we move forward. The Computer Science Academy is also a relatively new initiative, so we will need to refine the notification timetable, and how to best utilize the students that are in the buddy system to speak with students that are eligible for enrollment.

References

[1] S. Cheryan A. Master and A. N. Meltzoff. "Computing whether she belongs: Stereotypes undermine girls' interest and sense of belonging in computer science". In: *Journal of Educational Psychology* 108.3 (2016), pp. 424–437. DOI: `https://doi.org/10.1037/edu0000061`.

[2] E. Höhne and L. Zander. "Sources of male and female students' belonging uncertainty in the Computer Sciences." In: *Frontiers in Psychology* 10 (2019). DOI: `https://doi.org/10.3389/fpsyg.2019.01740`.

[3] Lockheed Martin. *CYBERQUEST® and Code Quest*. `https://www.lockheedmartin.com/en-us/news/features/2022/cyberquest-and-code-quest.html`. 2023.

[4] The State of New Jersey. *New Jersey State Minimum Graduation Requirements by Content Area 120 credits (N.J.A.C. 6A:8-5.1)*. `https://www.nj.gov/education/cccs/grad.pdf`. 2023.

[5] B. Kennedy R. Fry and C. Funk. *STEM Jobs See Uneven Progress in Increasing Gender, Racial and Ethnic Diversity*. `https://www.pewresearch.org/science/2021/04/01/stem-jobs-see-uneven-progress-in-increasing-gender-racial-and-ethnic-diversity`. 2021.

[6] South Brunswick High School. *Career Academies*. `https://sbhs.sbschools.org/school_information/s_b_h_s/career_academies`. 2023.

Regulating Generative AI: A Pathway to Ethical and Responsible Implementation*

Jonathan Luckett
College of Business, Innovation, Leadership, and Technology
Marymount University, Arlington, VA 22207
jpl65485@marymount.edu

Abstract

Artificial intelligence (AI) is becoming more and more prevalent in our daily lives, and its potential applications are practically limitless. However, as with any technology, there are concerns about how AI could be misused or abused. One of the most serious concerns is the potential for discrimination, particularly against women or minorities, when AI systems are used for tasks like job hiring. Additionally, there are concerns about privacy and security, as AI could be used to monitor people's movements or launch cyberattacks. To address these concerns, regulations must be developed to ensure that AI is developed and used ethically and responsibly. These regulations should address issues like safety, privacy, security, and discrimination. Finally, it is important to educate the public about AI and how to use it safely and responsibly. In this paper, I will examine the AI regulations and challenges that exist today, particularly in the United States. Two regulations I will focus on are the AI in Government Act of 2020 and the National Artificial Intelligence Initiative Act of 2020. Additionally, I will examine two Executive Orders that have addressed the issue of AI in the federal government. This paper examines AI generative tools, such as Bing, Bard, and ChatGPT. Finally, the paper concludes with some policy considerations and recommendations for federal agencies.

1 Introduction

As AI technology continues to rapidly advance and become increasingly integrated into our daily lives, it is crucial to consider the potential risks and concerns associated with its use. One of the most serious concerns is the potential for misuse, such as the development of autonomous weapons that could cause harm without human intervention. AI could also be used to discriminate against certain groups of people or to invade people's privacy, while also increasing the risk of cyberattacks on critical infrastructure. In addition to these specific concerns, there are also broader ethical concerns about AI, such as the possibility of machines surpassing human intelligence and becoming a threat to humanity, or the creation of a society controlled by AI. To mitigate these concerns, it is essential to develop regulations that ensure responsible and ethical development and use of AI, addressing safety, privacy, security, and discrimination issues. Additionally, investing in research and development of AI technologies that can be used to solve societal challenges and educating the public about AI's potential benefits and risks is crucial. There are several regulatory concerns that have been raised about AI. These concerns include:

- Safety: AI systems are complex and can be difficult to understand. This makes it difficult to ensure that they are safe to use. For example, AI systems used in self-driving cars could make mistakes that could lead to accidents. Privacy: AI systems collect and process large amounts of data. This data could be used to invade people's privacy. For example, AI systems could collect information about people's online activity and use it in ways that violate the individual's privacy.

- Security: AI systems could be hacked or used to launch cyberattacks. This could have a major impact on critical infrastructure, such as power grids and transportation systems. For example, an AI-powered cyberattack could shut down power grids or transportation systems, causing widespread chaos and disruption.

- Discrimination: AI systems could be biased, leading to discrimination against certain groups of people. For example, AI systems used for hiring could discriminate against women or minorities.

- Ethics: There are a number of ethical concerns about the development and use of AI. For example, some people worry that AI could become so intelligent that it surpasses human intelligence and becomes a threat to humanity. Others worry that AI could be used to create a society where people are controlled by machines.

This paper examines the AI regulations and challenges that exist today, particularly in the United States. Two regulations I will focus on are the AI in Government Act of 2020 and the National Artificial Intelligence Initiative Act of 2020. Additionally, I will examine two Executive Orders that have dealt with the issue of AI in the federal government. I will examine the rise of AI tools like Bing, Bard, and ChatGPT and conclude with some policy considerations and recommendations for federal agencies.

2 Policy Issues

A recent Bloomberg article discussed how the rapid advancement of generative AI, including chatbots that can create content on their own, is presenting new challenges for governments and regulators worldwide [1]. Potential issues include mass surveillance, creating inequities, and physical danger. The rapid advancement of generative AI, including chatbots that can create content on their own, is presenting new challenges for governments and regulators worldwide. The European Union has proposed regulations for AI in its Artificial Intelligence Act, which puts safeguards in place for high-risk applications while allowing for experimentation with lower-risk ones. The U.S. government has presented voluntary guidelines for an AI Bill of Rights, but experts argue that they do not address issues raised by generative AI, such as mass-produced disinformation. Some companies developing AI have been placing limits on themselves to ensure responsible development due to the lack of clear policies. However, there are concerns that overly stringent regulations may give China a geopolitical advantage in AI. China has already planned regulations to limit generative AI, potentially to implement censorship. As technologists push ahead with the development of generative AI, officials, and regulators may struggle to keep up [1].

High-risk applications of AI can include those that have significant potential for harm to individuals, society, or the environment. Some examples of high-risk applications of AI could include:

1. Healthcare: AI systems that assist in medical decision-making, diagnosis, and treatment planning have the potential to impact patient safety and outcomes. These systems must be carefully evaluated and tested to ensure they are safe and effective.

2. Autonomous vehicles: Self-driving cars and other autonomous vehicles rely heavily on AI to make decisions in real time. As these technologies become more prevalent, they will need to be regulated to ensure safety on the roads.

3. Financial services: AI is increasingly being used in financial services for tasks like fraud detection, risk management, and credit scoring. However, these systems must be monitored to prevent bias and ensure they are not used to perpetuate systemic inequalities.

4. Criminal justice: AI systems are being used in the criminal justice system for tasks like predictive policing, sentencing, and parole decisions. However, there are concerns about bias and the potential for these systems to worsen existing inequalities in the justice system.

5. Military applications: AI is being used for a variety of military applications, including autonomous weapons systems. There are concerns about the ethical implications of using AI in warfare and the potential for these systems to cause harm to civilians.

Many industries are worried about the possibility of job displacement due to the advancements in AI technology. The fear is that certain industries and job functions will be replaced by AI, particularly for tasks that are predictable, routine, and easily automated.

Numerous studies have been conducted to estimate the potential impact of AI on employment, but the results vary. For instance, McKinsey Company's 2017 report claims that up to 800 million jobs could be displaced globally by 2030 [2], while the World Economic Forum's 2018 report predicted that AI would create 133 million new jobs by 2022, but also displace 75 million [3].

It is no secret that many industries are concerned about the possible effects of AI on employment. The worry is that certain job functions and industries may be replaced by AI, particularly those that involve routine and predictable tasks that can easily be automated. While a variety of studies have been conducted to estimate the potential impact of AI on employment, the results vary. However, regardless of the exact numbers, it's clear that AI will have a significant impact on the labor market. Certain industries, like transportation and manufacturing, are particularly vulnerable to job displacement due to AI. For example, self-driving trucks and delivery drones could potentially replace millions of drivers. It's important to keep in mind, however, that AI will also create new job opportunities in other industries. Workers with skills in data science, machine learning, and AI engineering will be in demand. To decrease the potential impact of AI on job displacement, policymakers and industry leaders must work together to create new training and education programs to prepare workers for the jobs of the future. This includes retraining and upskilling workers in industries that are most vulnerable to job displacement, as well as investing in new education and training programs for emerging AI-related job roles. Additionally, policymakers must consider implementing policies such as job-sharing and shorter workweeks to reduce the negative impacts of AI-related job displacement on workers.

3 Executive Orders and the AI Bill of Rights

Executive Order 13859, signed by President Donald Trump on February 11, 2019, is titled "Maintaining American Leadership in Artificial Intelligence" and represents a major step forward in the United States' approach to regulating AI [4]. This order emphasizes the importance of promoting the development of AI and establishing the United States as a global leader in AI technology.

The order establishes AI as a key priority for the United States, recognizing that AI has the potential to drive economic growth and improve the quality of life for Americans. The order directs federal agencies to prioritize AI research and development in their budget proposals and to prioritize funding for AI-related programs.

The order also emphasizes the importance of public-private partnerships in advancing AI research and development. It calls for federal agencies to work with industry, academia, and other stakeholders to identify and address key challenges facing the AI industry.

One of the key elements of the order is the establishment of the American AI Initiative, which is designed to promote and protect American AI technology and innovation. The initiative includes five pillars: promoting AI research and development, creating a national AI workforce, establishing AI governance standards, developing international AI cooperation, and protecting America's AI advantage [2].

Under the first pillar, the order calls for federal agencies to prioritize AI research and development and to provide funding and support for AI initiatives. The second pillar focuses on developing the American AI workforce, including by investing in AI education and training programs and by promoting diversity and inclusivity in the AI industry.

The third pillar calls for the development of AI governance standards, including ethical and safety standards for the use of AI technology. The fourth pillar focuses on promoting international cooperation in AI research and development, including collaborating with international partners, and participating in international AI forums. Finally, the fifth pillar focuses on protecting America's AI advantage, including protecting intellectual property and ensuring that foreign entities do not gain access to American AI technology.

The executive order (EO 13960) signed by President Donald Trump on promoting the use of trustworthy artificial intelligence in the federal government was issued by the US government on December 8, 2020 [5]. It is intended to ensure that the government's use of AI is transparent, accountable, and consistent with American values.

The order establishes guidelines and requirements for the development and use of AI systems within the federal government. These guidelines include the importance of public participation in AI development, transparency in AI decision-making, and ethical considerations in the design and implementation of AI systems [5].

To support the implementation of these guidelines, the order establishes the AI Center of Excellence (AI CoE). The AI CoE will work with federal agencies to develop and use AI in a trustworthy and responsible manner, and to ensure that AI is used to enhance public trust and confidence in government services.

The executive order also emphasizes the importance of avoiding discrimination and bias in the use of AI. It requires agencies to develop plans for identifying and addressing potential sources of bias in AI systems and to provide training and resources to employees to ensure they understand and comply with the order's provisions [5].

Overall, the order is intended to ensure that the federal government uses AI responsibly and ethically and that the public has confidence in the government's use of this powerful technology.

The *Blueprint for an AI Bill of Rights* is a document released by the White House Office of Science and Technology Policy in October 2022 [6]. It outlines five principles and associated practices to help guide the design, use, and deployment of automated systems to protect the rights of the American public in the age of artificial intelligence. The five principles are:

1. Safe and Effective Systems. Automated systems should be designed and built to be safe and effective, and to avoid causing harm to people or the environment.

2. Algorithmic Discrimination Protections. Automated systems should be designed and built to avoid discrimination based on race, ethnicity, gender, sexual orientation, religion, national origin, or other protected characteristics.

3. Data Privacy. People should have the right to control their data, and to know how their data is being used by automated systems. Notice and Explanation. People should be given notice when their data is being used by an automated system and should be able to understand how their data is being used.

4. Human Alternatives, Consideration, and Fallback. Automated systems should not be used to make decisions that have a significant impact on people's lives without human oversight and consideration.

4 Key AI Framework

The regulation of artificial intelligence is a complex issue that is still evolving in the United States and abroad. While there is currently no comprehensive regulatory framework for AI in the US or globally, several initiatives have been undertaken to address the ethical, legal, and societal implications of AI. Here are some of the key developments in AI regulation in the US and abroad:

United States:

1. AI in Government Act of 2020: This law requires the Federal government to create a plan to facilitate the development and use of AI within the Federal Government [3].

2. National Artificial Intelligence Initiative Act of 2020: This law directs the President to implement a national strategy on AI research and development, with a focus on economic competitiveness, national security, and social and ethical considerations [4].

3. Executive Order on Maintaining American Leadership in Artificial Intelligence: This order directs Federal agencies to prioritize AI research and development and directs the National Institute of Standards and Technology to create standards for the development and use of trustworthy AI [7].

4. Executive Order on Promoting the Use of Trustworthy Artificial Intelligence in the Federal Government: This order directs Federal agencies to promote the use of AI that is safe, reliable, and transparent, and to develop standards and guidance for the use of AI in government [8]

5. Federal Trade Commission (FTC) guidelines: The FTC has issued guidelines on the use of AI in decision-making processes, emphasizing the need for transparency, fairness, and accuracy [5].

6. Various state initiatives: Several states have proposed or enacted legislation related to AI regulation, including California, which has passed a law requiring companies to disclose when they are using AI to generate content or manipulate audio or video recordings.

Europe:

1. General Data Protection Regulation (GDPR): The GDPR provides a legal framework for the protection of personal data in the European Union (EU), including data processed by AI systems.

2. European Commission's AI Regulation Proposal: In 2021, the European Commission proposed a new regulation on AI that would establish legal obligations for the development, deployment, and use of AI systems in the EU, with a focus on high-risk AI applications [9].

3. High-Level Expert Group on Artificial Intelligence: This group advises the European Commission on AI-related issues, including ethical and legal implications [10].

4. AI Ethics Guidelines: The European Union's AI Ethics Guidelines provide recommendations for the ethical development and deployment of AI, including the need for transparency, fairness, and accountability [7].

China:

1. The New Generation Artificial Intelligence Development Plan: This plan outlines China's strategy for becoming a world leader in AI by 2030, with a focus on developing key AI technologies and applications [11]

2. The Social Credit System: This system uses AI and other technologies to monitor and rate individuals' behavior and social status, raising concerns about privacy and freedom of expression [12]

3. The Cybersecurity Law: This law requires companies to disclose data breaches and establish data protection mechanisms, including those related to AI [13].

4. Various national initiatives: China has launched several initiatives related to AI regulation, including the establishment of a National AI Standardization Committee and a National Artificial Intelligence Open Innovation Platform.

Overall, while there is no comprehensive regulatory framework for AI in the U.S. or abroad, there are several initiatives and proposals underway to address the ethical, legal, and societal implications of AI. As AI continues to advance and become more integrated into our daily lives, it is likely that more regulatory measures will be implemented to ensure that its development and use align with societal values and interests.

The AI in Government Act of 2020 is a bill that was introduced in the United States Congress to promote the use of artificial intelligence technology in the federal government. The bill was enacted as Division U of the Consolidated Appropriations Act of 2021 [3].

The act has several key provisions aimed at promoting the use of AI in the government. It establishes a Federal Advisory Committee on AI, which is responsible for advising the federal government on issues related to the development and use of AI. The committee is also tasked with promoting the use of AI within the federal government and ensuring that AI is used in a manner that is consistent with the values of the United States.

The AI in Government Act also requires the General Services Administration (GSA) to establish a Center of Excellence for AI. This center is responsible for providing guidance and assistance to federal agencies on the use of AI, as well as promoting best practices and developing standards for the use of AI in the government [3].

In addition, the act requires federal agencies to develop plans for integrating AI into their operations. These plans must include a strategy for identifying and addressing any ethical and security issues that may arise from the use of AI. The plans must also address the need for training and education for federal employees on the use of AI.

The AI in Government Act also includes provisions related to the use of AI in procurement. It requires the GSA to establish a program for training federal acquisition personnel on the use of AI in procurement and to develop guidance on the use of AI in procurement. It also requires federal agencies to develop plans for the use of AI in procurement and to report to Congress on the use of AI in procurement.

Finally, the act establishes a pilot program for the use of AI in the federal government. The pilot program is designed to encourage federal agencies to develop and implement innovative uses of AI technology. The program is funded by appropriations from Congress and will be overseen by the Federal Advisory Committee on AI.

Overall, the AI in Government Act of 2020 is aimed at promoting the use of AI in the federal government. By establishing a Federal Advisory Committee on AI, a Center of Excellence for AI, and requirements for the development of plans for the use of AI, the act seeks to ensure that AI is used responsibly and effectively to improve government operations and services [3].

The National Artificial Intelligence Initiative Act of 2020, also known as Division E of the National Defense Authorization Act for Fiscal Year 2021, aims to advance the development and use of AI across various sectors in the United States. The bill recognizes the potential benefits that AI can bring to society and the economy but also acknowledges the ethical, legal, and social challenges that come with the deployment of this technology [4].

The bill establishes a National AI Initiative Office within the White House Office of Science and Technology Policy, which will be responsible for coordinating AI-related activities across government agencies and promoting research and development in AI. This centralization of efforts is expected to lead to a more cohesive and coordinated approach to AI development, which is necessary to ensure that the United States remains competitive in this field [4].

The bill also authorizes funding for AI research and development, including the creation of AI research institutes and the expansion of existing AI initiatives. It encourages the development of partnerships between government, industry, and academia to accelerate progress in AI research and deployment. The creation of AI research institutes is expected to bring together experts from different fields to work on specific AI-related issues, which is expected to lead to breakthroughs in this field.

In addition, the bill addresses the ethical and security concerns surrounding AI by requiring the development of guidelines for the responsible use of AI and promoting the integration of security measures into AI systems. It recognizes that the use of AI can raise ethical issues such as bias, privacy, and accountability, and requires that guidelines be developed to ensure that these issues are addressed. The integration of security measures is essential to prevent the misuse of AI systems, which could have serious consequences for society.

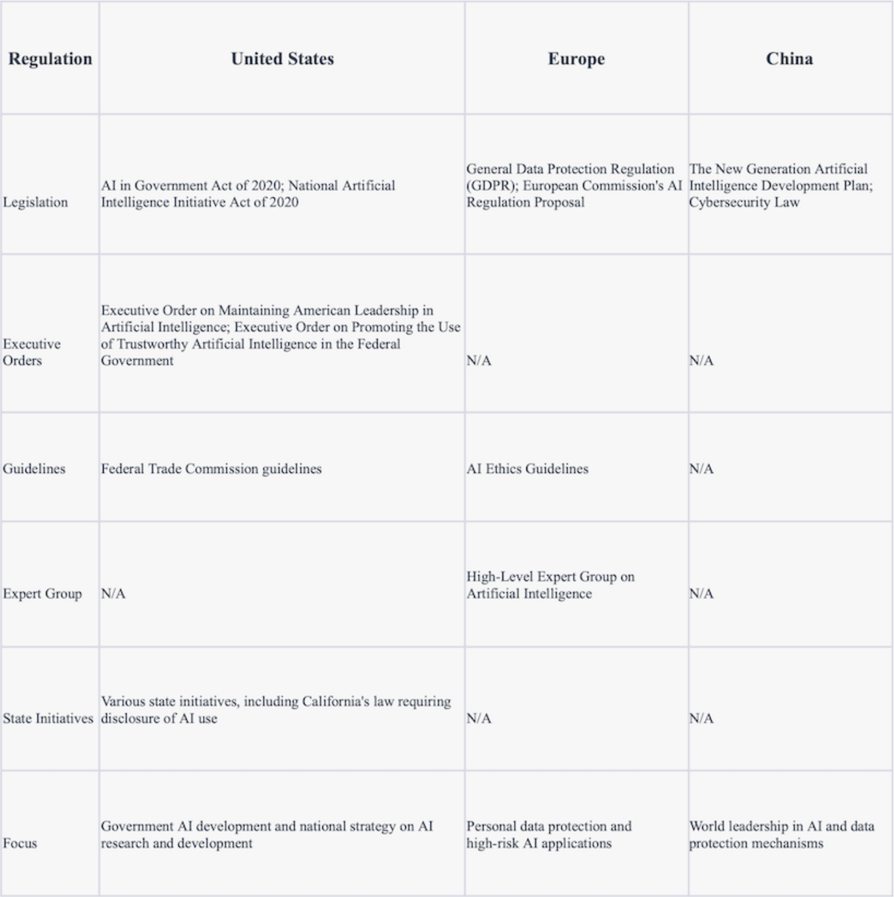
The bill also establishes a task force to examine the impact of AI on the workforce and make recommendations for workforce development programs. The task force is expected to examine the potential impact of AI on jobs, as well as the skills that will be required in the future economy. The recommendations of the task force will be important in ensuring that the workforce is adequately prepared for the changes that are expected to come with the deployment of AI [4].

Overall, the National Artificial Intelligence Initiative Act of 2020 represents a comprehensive effort to promote the development and responsible use of AI in the United States. It recognizes the potential benefits that AI can bring to society and the economy, but also acknowledges the challenges that come with its deployment. The establishment of the National AI Initiative Office, the authorization of funding for AI research and development, the development of guidelines for the responsible use of AI, and the examination of the impact of AI on the workforce are all important steps in ensuring that the United States remains competitive in this field while also ensuring that the deployment of AI is done responsibly and ethically. Table 1 displays the current regulatory framework for AI.

5 AI Generative Tool Challenges, Bing, Bard, and ChatGPT

AI is an AI generative tool from Microsoft; Bard is a large language model from Google AI; and ChatGPT is a large language model from OpenAI. ChatGPT and these other AI tools are one of the most advanced artificial intelligence technologies available today [9]. They can understand natural language, generate

Table 1: Current AI Regulatory Framework Comparison Chart



Regulation	United States	Europe	China
Legislation	AI in Government Act of 2020; National Artificial Intelligence Initiative Act of 2020	General Data Protection Regulation (GDPR); European Commission's AI Regulation Proposal	The New Generation Artificial Intelligence Development Plan; Cybersecurity Law
Executive Orders	Executive Order on Maintaining American Leadership in Artificial Intelligence; Executive Order on Promoting the Use of Trustworthy Artificial Intelligence in the Federal Government	N/A	N/A
Guidelines	Federal Trade Commission guidelines	AI Ethics Guidelines	N/A
Expert Group	N/A	High-Level Expert Group on Artificial Intelligence	N/A
State Initiatives	Various state initiatives, including California's law requiring disclosure of AI use	N/A	N/A
Focus	Government AI development and national strategy on AI research and development	Personal data protection and high-risk AI applications	World leadership in AI and data protection mechanisms

human-like responses, and complete various tasks like writing essays, composing music, and more. While Bing, Bard, and ChatGPT have tremendous potential to revolutionize many industries, they also pose significant challenges, especially in terms of its unregulated use.

One of the most significant challenges associated with these AI tools is their potential misuse. Being an AI model, Bing, Bard, and ChatGPT can be programmed to generate false or misleading information, promote harmful content, or even mimic human behavior to deceive individuals. The technology can be manipulated to spread disinformation, propaganda, and fake news. This poses a significant threat to the integrity of online content, which can be used to influence public opinion and interfere with democratic processes.

Another challenge with Bing, Bard, and ChatGPT are their potential impact on employment [12]. ChatGPT, for example, can be programmed to complete various tasks, including writing, editing, and proofreading. While this may seem like a positive development, it has the potential to automate various jobs currently performed by humans. This may lead to job losses in various industries, including journalism, content writing, and data entry. Additionally, these tool's ability to complete tasks quickly and accurately may lead to a decline in the quality of human work.

The potential misuse of Bing, Bard, and ChatGPT also raises significant ethical concerns. They can be programmed to learn from biased data, which can result in perpetuating existing social inequalities. For instance, a biased AI model can learn and reproduce existing racial or gender biases, leading to discriminatory outcomes. Additionally, these tools may pose a threat to personal privacy by collecting and storing personal data without proper consent or transparency.

Moreover, the lack of proper regulation and oversight is also a significant concern regarding Bing, Bard, and ChatGPT's unregulated use. Currently, there are no clear guidelines or standards on how to regulate the use of these tools or any other advanced AI technology. This leaves the door open for unethical practices, including using AI for political propaganda, hate speech, and other harmful activities.

The unregulated use of Bing, Bard, and ChatGPT poses significant challenges and ethical concerns. To mitigate these challenges, there needs to be increased regulation and oversight of AI technologies like ChatGPT. This includes the development of clear guidelines and standards for its use, ensuring that AI models are developed using unbiased data, and providing transparency regarding the collection and use of personal data. It is also essential to prioritize ethical considerations in the development and deployment of AI technologies to ensure they are used to benefit humanity and not cause harm.

It should be noted that there are numerous other AI tools that go beyond creating text and written word. For example, there is Dall-E 2, a tool that can create realistic images from text. There are voice generators that create human quality voice recordings. There are video generators that can create high-quality videos from text. And there are deep fake tools that create videos or audio of people saying and doing things that they have not done. These tools can spread misinformation, can stifle creativity, and invade individual privacy.

6 Policy Considerations

Several policy considerations and recommendations can be made to regulate AI, including Bing, Bard, and ChatGPT, and address the challenges identified above.

Firstly, it is important to establish clear legal frameworks and ethical guidelines for the development and deployment of AI. This should include requirements for transparency, accountability, and explainability of AI systems. Regulations should also ensure that AI systems do not discriminate against individuals or groups and protect the privacy of users.

Secondly, it is crucial to promote education and awareness about AI and its potential impacts. This includes providing training and education to individuals and organizations on the responsible use and development of AI, as well as increasing public awareness about AI and its potential impacts on society.

Thirdly, it is essential to establish cross-sector partnerships and collaboration to accelerate progress in AI research and development. This can involve collaboration between government, industry, and academia to share knowledge, resources, and expertise.

Fourthly, there is a need to focus on building an AI-ready workforce, which includes investing in education and training programs to equip workers with the skills needed to work alongside AI systems. This can involve the development of new curricula in schools and universities to focus on AI-related skills and training programs for existing workers.

Lastly, continuous monitoring and evaluation of AI systems should be implemented to ensure they remain in line with regulations and ethical guidelines. This includes establishing oversight bodies to monitor the development and deployment of AI systems and to investigate any breaches of regulations or ethical guidelines.

Overall, a comprehensive regulatory framework is required to ensure the responsible development and deployment of AI systems, including Bing, Bard, and ChatGPT. The policy recommendations outlined below can serve as a starting point for the development of such a framework, which should be continuously reviewed and updated as AI technology advances and new challenges emerge.

7 Proposed Legislation in the United States

Below are some specific proposed regulations or legislation for AI in the United States including the Algorithmic Accountability Act, the Artificial Intelligence Non-Discrimination Act, and the Facial Recognition and Biometric Technology Moratorium Act.

The Algorithmic Accountability Act would require companies to disclose how their algorithms work and to take steps to mitigate bias. The act would also establish a new office within the Federal Trade Commission (FTC) to oversee the use of algorithms. The act was introduced in the House of Representatives in 2021 by Representatives Yvette D. Clarke (D-NY), Raja Krishnamoorthi (D-IL), and Jim Himes (D-CT). It has been referred to the House Committee on Energy and Commerce but has not yet been scheduled for a vote [14].

The Facial Recognition and Biometric Technology Moratorium Act would place a moratorium on the use of facial recognition technology by the federal government until certain privacy and civil liberties concerns are addressed. The act would also establish a new commission to study the use of facial recognition technology and make recommendations for its regulation. The act was introduced in the House of Representatives in 2020 by Representatives Rashida Tlaib (D-MI), Pramila Jayapal (D-WA), and Ayanna Pressley (D-MA). It was passed by the House of Representatives in 2021 but has not yet been taken up by the Senate [15].

In 2020, the FTC issued a call for public comment on proposed changes to the agency's rules to require that companies that use AI explain the algorithms and decision-making processes they use in consumer-facing applications, such as chatbots or predictive analytics. The FTC's proposed rulemaking would require companies that use AI to explain the algorithms and decision-making processes they use in consumer-facing applications. This would give consumers more information about how AI is being used to make decisions about them and would help to prevent discrimination and other harms [5].

The proposed rulemaking would apply to a wide range of companies, including those that use AI for things like targeted advertising, credit scoring, and hiring decisions. The rule would require companies to provide consumers with information about how AI is used, including the data that is used to train the algorithms, the factors that are considered in making decisions, and the accuracy of the algorithms.

The FTC's proposed rulemaking is supported by a number of consumer groups and privacy advocates. They argue that the rule would help to protect consumers from discrimination and other harms and would give them more control over their personal data. However, the proposed rulemaking is opposed by some businesses. They argue that the rule would be too costly and burdensome and would stifle innovation.

The Federal Trade Commission proposed rulemaking on algorithmic transparency is still in progress. The FTC received over 4,500 comments on the proposed rule and is currently in the process of reviewing and analyzing those comments. The FTC has not yet announced a timeline for finalizing the rule.

NIST's draft AI Risk Management Framework provides guidance for organizations on how to identify and mitigate potential risks associated with AI. The framework covers a wide range of risks, including bias, security vulnerabilities, and privacy violations. NIST's draft AI Risk Management Framework is also still in progress. NIST is currently seeking feedback on the draft framework from the public. The framework is expected to be finalized in late 2023 or early 2024 [16].

Both the FTC's proposed rulemaking and NIST's draft framework are important steps in ensuring that AI is developed and used in a responsible and ethical manner. The frameworks provide guidance for organizations on how to identify and mitigate potential risks associated with AI, such as bias or security vulnerabilities. These frameworks will help to ensure that AI is used for good and does not harm consumers or society.

8 Federal Policy Recommendation

Federal agencies need specific policies to deal with content that is developed by AI tools. Additionally, these agencies also require policies that address safety, privacy, and ethical issues. It is recommended that step 1 is for federal agencies to publish a policy that addresses the creation and publishing of content by AI tools.

Policy: All federal agencies that produce content and publish documents, presentations, guidance, standards, frameworks, models, and any other type of publishable content will be required to cite any generative AI tools or algorithms used in the creation of such content. The citation should include the name of the AI tool or algorithm, the date of use, and any other relevant information.

Rationale: The use of generative AI in the creation of content has become increasingly common across various industries, including in the production of government-related documents and materials. As such, federal agencies need to acknowledge the use of generative AI in the creation of any publishable content. The following are the reasons why this policy is necessary:

- Attribution: Federal agencies have a responsibility to provide accurate and transparent information to the public. Citing the use of generative AI in the creation of content is important for attributing credit and acknowledging the role that AI played in the creation of the content.

- Transparency: The use of generative AI in the creation of content can be controversial, especially when it comes to government-related materials. Citing the use of generative AI helps to provide transparency and clarity about the creation process and can help to dispel any concerns about the authenticity or originality of the content.

- Accountability: By requiring federal agencies to cite the use of generative AI in the creation of publishable content, it promotes accountability and responsibility in the use of these technologies. It helps to ensure that AI is being used ethically and thoughtfully in government-related materials and that the creators are aware of the potential implications of using AI in the content creation process.

- Education: Encouraging federal agencies to cite the use of generative AI in their content creation can also help to promote education and awareness about these tools and algorithms. This can lead to a better understanding of the capabilities and limitations of AI and can help to foster a more informed and thoughtful approach to using these tools in government-related materials.

Overall, requiring federal agencies to cite any generative AI tools or algorithms used in the creation of publishable content can help to promote attribution, transparency, accountability, and education. By acknowledging the role that AI plays in the content creation process, the agencies can help to ensure that it is being used thoughtfully and responsibly in government-related materials.

9 Regulatory Recommendations

Below are specific recommendations for addressing the regulatory framework concerns cited above in this paper:

1. Safety: AI systems must be designed and developed with safety in mind. This includes conducting thorough testing and validation to ensure that AI systems are reliable and free from errors that could harm users. Regulatory agencies should establish safety standards for AI systems, and companies that develop or use AI should be required to comply with these standards.

2. Privacy: AI systems can collect and process vast amounts of personal data, which can raise significant privacy concerns. Regulatory agencies

should establish clear guidelines and regulations for the collection and use of personal data in AI systems, including requirements for obtaining user consent, data transparency, and data security.

3. Security: As with any technology, AI systems can be vulnerable to cybersecurity threats. Regulatory agencies should establish cybersecurity standards for AI systems, and companies that develop or use AI should be required to comply with these standards. Additionally, companies should be required to implement robust security measures to protect against cyber attacks and data breaches.

4. Ethical considerations: AI systems can raise a number of ethical concerns, including issues related to bias, discrimination, and accountability. Regulatory agencies should establish ethical guidelines for the development and use of AI systems, with a focus on transparency, fairness, and accountability. Companies that develop or use AI should be required to comply with these guidelines.

5. Education and awareness: As AI technology continues to evolve, it is important that policymakers, industry leaders, and the public are educated about the potential benefits and risks of AI. Regulatory agencies should establish educational programs and awareness campaigns to promote a better understanding of AI and its implications.

6. Collaboration: The development and regulation of AI should involve collaboration between government agencies, industry leaders, academic institutions, and other stakeholders. This can help to ensure that AI systems are developed and regulated in a way that is responsible, ethical, and aligned with societal values.

7. Regular evaluation and revision: The regulatory framework for AI should be regularly evaluated and revised as necessary to keep pace with technological advancements and changing societal values. This can help to ensure that the regulation of AI remains effective and relevant over time.

10 Conclusion

Artificial intelligence has the potential to revolutionize various industries, but it also poses several challenges that need to be addressed. As AI technology continues to advance, it is imperative to establish appropriate regulations and guidelines to ensure its responsible use.

The current regulatory landscape around AI is still in its infancy, with various governments taking steps to address the challenges posed by AI. The

United States, for instance, has enacted several regulations aimed at promoting the use of AI while also ensuring its responsible use. However, there is still a long way to go in terms of establishing a comprehensive regulatory framework that addresses all the challenges posed by AI.

The challenges posed by AI range from ethical concerns such as privacy, fairness, and bias, to technical concerns such as security and transparency. These challenges require careful consideration and collaboration between various stakeholders, including governments, industry, academia, and civil society.

In conclusion, the regulation of artificial intelligence is a complex and multifaceted issue that requires careful consideration of the ethical, legal, and societal implications of AI. While there is currently no comprehensive regulatory framework for AI in the United States or globally, several initiatives have been undertaken to address these concerns.

The AI in Government Act of 2020 and the National Artificial Intelligence Initiative Act of 2020 are significant steps in promoting the development and use of AI in a responsible, ethical, and aligned manner. The Executive Orders on Maintaining American Leadership in Artificial Intelligence and Promoting the Use of Trustworthy Artificial Intelligence in the Federal Government provide additional guidance to Federal agencies on prioritizing and promoting the use of trustworthy AI. The FTC guidelines and various state initiatives also contribute to the growing awareness of the need for transparency, fairness, and accuracy in the use of AI.

The European Union's GDPR, AI Ethics Guidelines, and the proposed AI Regulation are important developments in promoting the ethical and responsible development and deployment of AI in Europe. China's New Generation Artificial Intelligence Development Plan, Cybersecurity Law, and various national initiatives demonstrate China's ambitious goals to become a leader in AI while also addressing concerns about privacy and data protection.

To further address the challenges and concerns related to AI, policymakers and industry leaders must work collaboratively to develop a comprehensive regulatory framework that promotes innovation while also ensuring the responsible and ethical development and use of AI. This framework should prioritize transparency, fairness, accountability, and privacy, and address concerns related to bias, safety, and security. In addition, efforts should be made to promote education and public awareness about the benefits and risks of AI to ensure that the public is informed and engaged in the development and use of AI.

Overall, AI has the potential to revolutionize our lives in countless ways, but it also presents significant challenges and concerns. As such, it is crucial that we continue to prioritize the responsible and ethical development and use of AI to ensure that it serves the best interests of society.

References

[1] Lucy Papachristou and Jillian Deutsch. ChatGPT leaves governments scrambling for AI regulations. 3 2023.

[2] Mckinsey Company. JOBS LOST, JOBS GAINED: WORKFORCE TRANSITIONS IN A TIME OF AUTOMATION. *https://www.mckinsey.com/ /media/BAB489A30B724BECB5DEDC41E9BB9FAC.ashx*, 12 2017.

[3] J McNerney. H.R.2575 - AI in Government Act of 2020, 12 2019.

[4] EB Johnson. H.R.6216 - 116th Congress (2019-2020): National Artificial Intelligence Initiative Act of 2020, 3 2020.

[5] Using artificial intelligence and algorithms, 6 2022.

[6] The White House. Blueprint for an AI Bill of Rights | OSTP | The White House, 3 2023.

[7] Maintaining American leadership in artificial intelligence, 2 2019.

[8] Executive Order on Promoting the Use of Trustworthy Artificial Intelligence in the Federal Government – The White House, 12 2020.

[9] EUR-LEX - 52021PC0206 - EN - EUR-LEX.

[10] High-level expert group on artificial intelligence, 6 2023.

[11] Full translation: China's 'New Generation Artificial Intelligence Development Plan' (2017) - DigiChina, 10 2021.

[12] Drew Donnelly, PhD. China Social Credit System Explained - How it works [2023]. *Horizons*, 4 2023.

[13] R Creemers, G Webster, and P Triolo. Translation: Cybersecurity Law of the People's Republic of China (Effective June 1, 2017) - DigiChina, 8 2022.

[14] YD Clarke. Text - H.R.6580 - 117th Congress (2021-2022): Algorithmic Accountability Act of 2022, 2 2022.

[15] P Jayapal. H.R.3907 - 117th Congress (2021-2022): Facial Recognition and Biometric Technology Moratorium Act of 2021, 6 2021.

[16] Elham Tabassi. Artificial Intelligence Risk Management Framework (AI RMF 1.0) | NIST. *NIST*, 1 2023.

Plant Disease Detection and Classification Using Deep Learning Models[*]

S. Konduru[1], M. Amiruzzaman[1], V. Avina[1], M. R. Islam[2]
[1]Department of Computer Science
West Chester University
West Chester, PA 19383, USA
{sk984965,mamiruzzaman,va985361}@wcupa.edu
[2]Department of Computer Science and Engineering
Bangladesh University of Business and Technology
Rupnagar, Mirpur-2, Dhaka-1216, Bangladesh
md.rajibul.islam@bubt.edu.bd

Abstract

Detection of diseases in plants at an early stage is crucial to achieving high yields, preserving crop quality, and effective disease management. Existing research focuses mostly on leaf disease detection, despite the fact that disease may develop everywhere on the plant. We developed a new dataset using the PlantVillage dataset and other online sources. We used Convolutional Neural Network (CNN) architectures, Alexnet and MobileNet to analyze and evaluate the performance of the models on the new dataset (i.e., consists of over 50,000 images). The models were trained on the new dataset for 100 epochs. MobileNet outperformed the other two models, attaining 99.69% training accuracy, 94.37% validation accuracy, 96% average precision, 96% recall, and an F1-score. The MobileNet model predicted diseases that affect portions of the plant other than the leaf better. This work demonstrates detecting plant disease and provides a feasible technique for enhancing crop management.

1 Introduction

Agriculture is an essential and underrated sector that provides us with our major source of food. Because of the expanding population, there is a need for increased food production. In the last decade, many agricultural fields have been transformed into commercial lands because of urbanization, perhaps leading to food scarcity. Crop production must be increased to meet food requirements over the increasing population. Pests and diseases are two of the most significant elements influencing crop yield [5]. It is critical to diagnose ailments early and treat them effectively to obtain high yields. This also helps to retain the quality and quantity of the food [3].

Other benefits include low-cost treatment, preventing the disease from spreading further, etc. Overall, it is necessary to detect plant diseases for increased food production, food security, and environmental protection [2, 3]. The most common cause of plant diseases is pathogens such as bacteria, fungi, viruses, and nematodes. These pathogens feed on plants and absorb nutrients from the plant, shrinking the growth and further damaging the leaves, stems, and roots, leading to a systematic infection that spreads through the entire plant resulting in death.

Following the guidelines and experience from existing work, the proposed work uses Convolutional Neural Network based algorithms such as MobileNet and AlexNet for the detection and classification of plant diseases. These models seems to have better performance in classifying and detecting plant-based diseases [1, 4, 7]. The work focuses on detecting diseases that occur on other parts of the plant besides leaf. To detect the disease at early stage, it is necessary to diagnose the entire plant, not just the image. The proposed model is intended to consider the whole plant for the diagnosis. Due to the lack of data on other parts of the Apple crop, data augmentation was performed by collecting the images from google. This study only focused on detection and identification of the plant diseases not the source of the diseases and economic aspects.

Three different crops, such as apple, tomato, and corn, are considered for training the model. Tomatoes and corn have only leaf images; we tried to cover most of the other parts of apples. A dataset containing 25 distinct types of diseases (classes) associated with apples, corn, and tomatoes is considered. The images that belong to leaf diseases are extracted from the Plant Village dataset, while the images diseases that occurs on stems, roots and fruits are augmented from the images collected from the web. The training dataset consists of 40,000 images while the testing dataset consists of 10,000 images. The main objective is to analyze the performance of the models on the newly created dataset. Once these models are trained, their performance is compared based on accuracy and latency.

Some of the contributions of the proposed work include:

1. Data balancing using under-sampling and oversampling to balance the class distribution in training data. Various data augmenting techniques are used to generate more images from the limited existing images through geometric and color transformations.

2. The model is trained to detect diseases that occur on all parts of the plant, not just the leaves. There are no datasets available on diseases that occur on stems, fruits, or roots. The current datasets only detect leaf disease, which is why we generated a dataset by gathering and augmenting relevant images from Google and other sources. This dataset contains data that was acquired in an uncontrolled environment.

3. Multiple CNN models are trained and tested to see how they learn and perform on the new dataset.

2 Related work

We studied some related work and they helped us to gain valuable insights and what are the shortcomings in the existing works. A synopsis of the related works are presented below:

In [4], Mahmudul et al. replaced the standard convolution method to reduce the computation cost and the number of parameters. The work focused on leaf diseases in 14 different plant species. Similarly, Sasikala et al. [11] also used different CNN models, such as, ResNet 50 & 101, InceptionV3, DenseNet121 & 201, MobileNet V3, and NasNet. To overcome overfitting of the model, various data augmentation techniques such as image enhancement, scaling, rotation, and translation were applied. The Plant-Village dataset, that consists of around 38 classes belonging to 14 different types of crops was used to train the pretrained models.In [7], Swathi et al. proposed a model called AgriDoc for the detection and classification of plant leaf diseases. The Plant Village dataset was used, which consists of 38 different classes. Different convolutional neural network models, such as AlexNet, MobileNet, ResNet and some built from scratch CNN-based models were implemented and compared based on their performance and accuracy.

In [1], Rinu et al. used VGG16, which is a CNN based model for the detection of plant leaf diseases belonging to 38 classes. In [2], Chowdhary et al. achieved 98% accuracy on average using U-net models. Furthermore, the authors achieved 99% accuracy as they used EfficientNet-B4 model. The work showed that the proposed models performed better when they were trained on segmented images with deeper networks. It was mentioned that the proposed work outperformed the existing work for the same purpose. Not as high accuracy as [1] and [2]. But, Nishant et al. [10] achieved 95.6% accuracy based on a CNN-based model (i.e., VGG19) and detected and classified plant leaf diseases that belong to 13 different species. Arun Pandian et.al [9] achieved 99.79% accuracy on average. In [8], Lili Li et al. reviewed different works on plant disease detection and classification using deep learning techniques in the past year.

In [3], Haridasan et al. proposed a system that detects rice crop diseases using computer vision, image processing, machine learning, and deep learning techniques. The authors aimed to achieve efficiency and reduce losses in the farming industry. Some of the most common rice diseases, such as brown leaf spot, rice blast, sheath rot, bacterial leaf blight, and false smut, are considered for training the model. Advanced techniques like image segmentation, support vector machine classifier, and convolutional neural network algorithms have been employed to detect and classify diseases accurately. The CNN method achieved better results than SVM in terms of accuracy. The model achieved 91.45% accuracy with ReLu and Softmax activation functions.

After studying and analysing related works, we can say that the existing works focused only on detecting leaf diseases while ignoring that disease can occur on any part of the plant, including root, stem, fruit, etc. Most of the existing works used the Plant Village dataset, which has a class imbalance problem. Some of the classes in the dataset have a greater number of images, while other classes have fewer images. If there is a huge imbalance between the classes, the model prioritizes in learning more from majority class thus resulting in huge error rate in predicting from minority class. Model will be biased and over-predicts the majority class and under-predicts the minority class further resulting in poor accuracy. Existing works have trained models on datasets with limited scope of diseases.

3 Methodology

The methodology for plant disease detection involves multiple phases that include (see Figure 1):

3.1 Data Collection

The images that belong to Apple, Corn and Tomato leaf diseases are acquired from the existing dataset, Plant Village while the images that belong to the diseases that occur on other plant parts are acquired from google. Due to the lack of images related to certain diseases, only 50-100 images that belong to 9 different classes were downloaded and labelled properly. The data from these 9 classes were balanced through oversampling using data augmentation

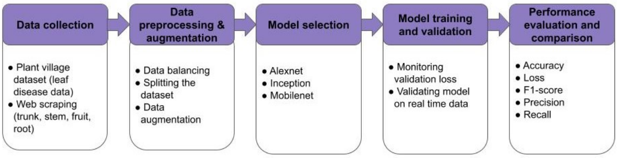

Data collection
• Plant village dataset (leaf disease data)
• Web scraping (trunk, stem, fruit, root)
Data preprocessing & augmentation
• Data balancing
• Splitting the dataset
• Data augmentation
Model selection
• Alexnet
• Inception
• Mobilenet
Model training and validation
• Monitoring validation loss
• Validating model on real time data
Performance evaluation and comparison
• Accuracy
• Loss
• F1-score
• Precision
• Recall

Figure 1: Overview of Proposed Methodology

techniques which will be discussed in detail in the further section. The images that belong to 16 classes were balanced through under-sampling by using splitting function. Overall, the dataset contains 50,000 images out of which 40,000 images were used for training the model and 10,000 images were used for validating the model. Out of 25 classes, each class contains 2000 images before they were split into training and testing datasets.

3.2 Data Pre-processing and Augmentation

The images that are downloaded from google are carefully examined and the images that have a lot of noise are removed. Once the images that belong to different classes are properly selected, they are further augmented using different augmentation techniques (see Figure 2):

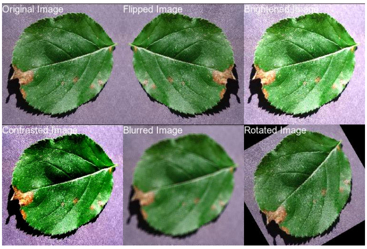


Figure 2: Illustration of different transformations applied to the original image.

Data augmentation is a method used to increase the size and diversity of the existing data by applying either geometric or color transformations. Geometric transformations include flipping, cropping, rotating (0 to 360), zooming and scaling of an image to generate new images while the color transformations include brightness and contrast. These transformations help the model to generalize unseen data and reduce over-fitting.

3.3 Model Selection

The main focus of the proposed work is not to achieve high accuracy using a particular deep learning model; it is to analyze how the deep learning models are performing on the augmented data that belongs to the whole plant and not just leaf. In this work, AlexNet and MobileNet models are chosen as they were proven to have achieved standard and consistent results in many state of the art approaches. Also, MobileNet consumes less computational power and less execution time.

3.3.1 AlexNet

AlexNet is one of the signifant convolutional neural networks developed in 2012 [6], by Alex krizhevsky, Ilya Sutskever and Geoffrey Hinton. This architecture consists of 8 layers in which the first 5 layers are convolutional layers and the remaining three are fully connected layers.

3.3.2 MobileNet

MobileNet is also a convolutional neural network which is known for its ease of computation compared to other CNN architectures. It contains multiple convolutional layers that are followed by average pooling layer and a SoftMax output layer that is fully connected. Image pre-processing and normalization is performed in the first layer. This architecture consists of two different convolutions such as depthwise and pointwise convolution.

3.4 Experimental Setup

Deep learning models typically require significant computational resources to train, especially for larger and more complex models or datasets. The dataset that is used for training has over 50,000 images.

3.5 Model Training

The training and validation results for each model is discussed below.

3.5.1 AlexNet

The model was trained on augmented dataset for 100 epochs with batch size 32 monitoring the loss. The model ran for 30 hours continuously.

The model achieved 99.19% training accuracy and 91.58% validation accuracy at epoch 100 whereas the training loss was 0.0286 and the validation loss was 2.6838 at epoch 100. The accuracy with the best loss value is 88.70%

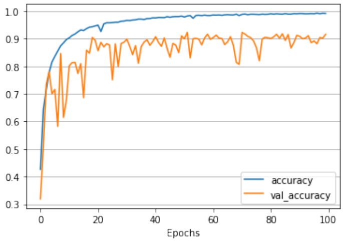


Figure 3: Training vs Validation accuracy of AlexNet model

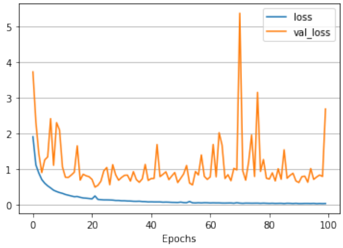


Figure 4: Training vs Validation loss of AlexNet model

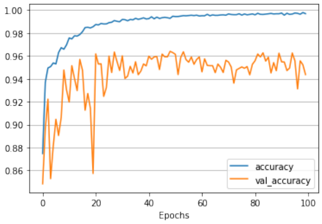


Figure 5: Training vs Validation accuracy of MobileNet model

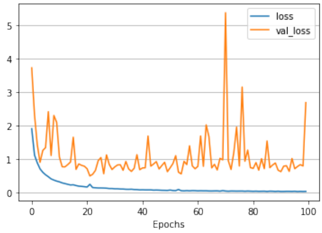


Figure 6: Training vs Validation loss of MobileNet model

(see Figure 4). The model is evaluated in terms of accuracy, F1 score, loss, precision, and recall.The trained model weights are saved as a h5 file for future use and the model data is saved to CSV format for data analysis.

3.5.2 MobileNet

The model was trained on augmented dataset for 100 epochs with batch size 32. Pre-trained imagenet weights were used and the layers were not re-trained on the new dataset. The model was initially trained by retraining some of the layer on the augmented dataset, but the results showed overfitting because of which we kept experimenting with layers retraining but the results were no so prominent either of the cases. Finally, we tried to keep the Imagenet weights as it is without retraining the layers and the results looked better as well. The accuaracy and the loss can be seen in (5) and 6).

The evaluation is performed in terms of accuracy, precision, loss and recall.

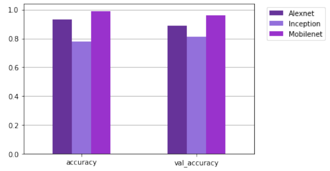


Figure 7: Comparing training & validation accuracy of the models

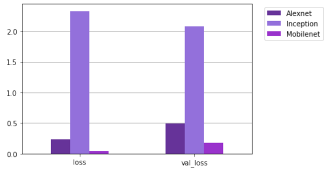


Figure 8: Comparing training & validation loss of the models

3.6 Performance Evaluation

The performance of the models AlexNet and MobileNet are compared in this section in terms of accuracy, loss, precision and recall.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1\ Score = 2 \times \frac{(precision \times recall)}{(precision + recall)} \tag{4}$$

TP represents true positives, TN represents true negatives, FP represents false positives, and FN represents false negatives.

The training and validation loss of the models are compared and the results are shown in the Figure 8).The loss of AlexNet is better but it is still high compared to MobileNet. Overall, Mobilenet performed better with lower loss compared to other two models.

Table 1 shows the comparison of models using performance metrics such as accuracy, precision, recall and F1-score. The accuracy values mentioned in Table 1 are the values taken for best validation loss value. The results clearly show the superiority of MobileNet architecture over the other two models. Also, MobileNet took less time for training compared to other two models because of the less number of parameters. The results are not compared with the state of the art approaches as they are trained on data that only has leaf disease

Table 1: Performance Evaluation of CNN models

Model	Accuracy	Precision	Recall	F1 Score
AlexNet	88.70%	0.89	0.89	0.89
MobileNet	95.99%	0.96	0.96	0.96

images. Since this work focuses on diseases that occur on all parts of the plant, it is not efficient to compare the results with the existing methods.

4 Conclusion

Deep learning is one of the advanced technologies that is being implemented in various domains, including agriculture. In this study, we worked on disease detection. In addition, we have created our dataset by collecting images from various sources and augmenting the images to make sure all the classes are balanced and has over 50,000 images that belong to 25 different classes. Only three crops, including apples, corn, and tomatoes are considered for the simplicity purpose.

Convolutional neural network architectures such as AlexNet and MobileNet are considered after thorough research of the existing works. These models are trained on the proposed dataset to analyze and evaluated in terms of accuracy, precision, recall, and F1 score. As per the results, MobileNet performed better than AlexNet model. Also, it was faster in training and predicting disease detection.

As for the future study, real-time photos and augmented images can be added to the dataset to increase the model's performance. Also, using an image translation method, the Cycle GAN architecture may be utilized to expand the dataset.

References

[1] SP Adarsha et al. "Identification of Plant Diseases Based on Lesion Spots". In: *Journal homepage: www. ijrpr. com ISSN* 2582 (2022), p. 7421.

[2] Muhammad EH Chowdhury et al. "Automatic and reliable leaf disease detection using deep learning techniques". In: *AgriEngineering* 3.2 (2021), pp. 294–312.

[3] Amritha Haridasan, Jeena Thomas, and Ebin Deni Raj. "Deep learning system for paddy plant disease detection and classification". In: *Environmental Monitoring and Assessment* 195.1 (2023), p. 120.

[4] Sk Mahmudul Hassan et al. "Identification of plant-leaf diseases using CNN and transfer-learning approach". In: *Electronics* 10.12 (2021), p. 1388.

[5] Md Rajibul Islam et al. "An Efficient Technique for Recognizing Tomato Leaf Disease Based on the Most Effective Deep CNN Hyperparameters". In: *Annals of Emerging Technologies in Computing (AETiC)* 7.1 (2023).

[6] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems 25 (NIPS'2012)*. 2012.

[7] H Kushal et al. "AgriDoc: Classification and Prediction of plant leaf diseases". In: *Turkish Journal of Computer and Mathematics Education (TURCOMAT)* 12.14 (2021), pp. 3909–3932.

[8] Lili Li, Shujuan Zhang, and Bin Wang. "Plant disease detection and classification by deep learning—a review". In: *IEEE Access* 9 (2021), pp. 56683–56698.

[9] J Arun Pandian et al. "Plant disease detection using deep convolutional neural network". In: *Applied Sciences* 12.14 (2022), p. 6982.

[10] Nishant Shelar et al. "Plant Disease Detection Using Cnn". In: *ITM Web of Conferences*. Vol. 44. EDP Sciences. 2022, p. 03049.

[11] Sasikala Vallabhajosyula, Venkatramaphanikumar Sistla, and Venkata Krishna Kishore Kolli. "Transfer learning-based deep ensemble neural network for plant leaf disease detection". In: *Journal of Plant Diseases and Protection* 129.3 (2022), pp. 545–558.

Common C++ Pointer Misconceptions: Essence and Treatment*

James A. Jerkins[1] Yasmeen Rawajfih[2] Daniel A. Ray[1]
[1]Computer Science and Information Systems
University of North Alabama, Florence, AL 35630
{jajerkins,dray4}@una.edu
[2]Computer Science
Tuskegee University, Tuskegee, AL 36088
yrawajfih@tuskegee.edu

Abstract

This paper follows the evolution of persistent pointer and memory management misconceptions through the common CS undergraduate curriculum. The authors surveyed a diverse sample of undergraduate students from two different institutions of higher learning; both housing mature, ABET accredited, undergraduate computer science degree programs. This paper investigates the nature of underlying ill-informed notions about pointers and related topics across the CS curriculum. Finally, the authors provide suggestions for improved curriculum and pedagogy based on the findings.

1 Introduction

The dual concepts of pointers and dynamic memory are foundational concepts in C and C++ programming. Learning how to use basic pointers and allocate/deallocate memory is a critical part of CS curriculum [6]. Students' difficulties with pointers are widely acknowledged by CS educators and well documented. Researchers have investigated the common types of mistakes that learners make and devised pedagogical tools to assist students [1, 4, 8].

The literature surrounding common misconceptions around pointers and memory management clearly identify that not only are pointers a major hurdle for student learning (identified as a core transformative idea by Boustedt et al. [3]), but also that a student's "working" mental model around pointers often only supports usage in a very limited scope [4].

Adcock et al. constructed a model and taxonomy of pointer manipulations to investigate common pointer mistakes by students in a CS2 course [1]. They observed the three most common types of errors CS2 students make are "creating a memory leak by pointer leaving scope", "dereferencing dead pointer by using * or ->", and "creating a memory leak by using = (i.e. assignment)". Craig and Petersen created a taxonomy of key pointer concepts and ranked them by difficulty [4].

Qjan and Lehman presented a meta-analysis of the common misconceptions and challenges beginning students face when learning to program in general [8]. They highlight struggles students have with conceptual and strategic knowledge.

Allevato et al. designed a tool for students to use when debugging pointer issues called Dereferee [2]. The Dereferee tool is based on the checked pointers idea and the Checkmate tool [7]. Allevato et al. argue that the inability to easily determine why pointer code fails encourages students to abandon reasoned exploration of the problem and resort to "caveman debugging" (sprinkling print statements everywhere).

Freedman identifies several "oddities of C++" which include the atypical usage of char* pointers in certain contexts, the overloaded behavior of the stream insertion operator («), and others [5]. It is precisely these kinds of oddities that this study leverages to measure student understanding (or lack thereof).

This study investigates the maturity of student mental models by examining explanations about the use of pointers, operator overloading, the data type concept, and literal strings in a small C++ program. The authors hypothesize that an initial shallow conception persists and inhibits student growth.

2 Methodology

To test the hypothesis the authors presented a representative sample of CS undergraduate students a small C++ program involving basic use of pointers and string literals and asked them to explain, in detail, the behavior of the machine and the state of memory as the program progressed. The code and associated questions, discussed in Section 2.1, were designed to uncover students' understanding of both foundational and advanced CS topics like: the separate notions of pointer declaration and memory allocation, using pointers to reference C-strings (generated from string literals), the immutability of string literals, the relationship between pointers and arrays, and the context specific nature of operators (i.e. how operator behavior changes based on the operand or operands).

Responses were collected from two separate institutions at two distinct points of curricular progression. The first population, hereafter referred to as the intermediate population (IP), was drawn from classes where pointers are first introduced (CS2 courses). Among this group a basic understanding of the examined topics was expected, but not a mature mental model. A second population, the study population (SP), was sampled from senior-level courses. This approach permitted a comparative analysis between the two sample sets.

The first population of students (IP; intermediate population) was composed of two sections of a sophomore-level CS2 programming class (n=26) at institution A. The second population (SP; study population) was composed of single sections of two different courses available to senior-level students. The first was a network programming course (n=17) at institution A, and the second was an ethics course (n=15) at institution B. Both institution A and institution B are regional institutions of higher learning that each house mature, ABET accredited, undergraduate computer science degree programs.

2.1 Student Survey Instrument

The authors presented students with the printed C++ program shown in Listing 1. Along with this extremely simple program students were asked a series of questions (three questions for IP, four questions for SP) that required them to examine and mentally execute the program. Responses were written by hand and submitted to the respective investigators.

Listing 1: Assessment program

```
#include <iostream>
using namespace std;

int main()
{
    char* p;

    p = "some␣text";

    cout << *p << endl;
    cout << p << endl;

    return 0;
}
```

The questions were slightly modified for IP and SP to better fit the specific population. The questions and each specific variation are presented below:

- Question 1.(IP version) Explain what happens in memory when line #6 is executed.
- Question 1.(SP version) Explain what happens when the computer executes line #6; including how memory is affected.
- Question 2. (IP version) Explain what happens when line #8 is executed.
- Question 2. (SP version) Explain what happens when the computer executes line #8 including how memory is affected.
- Question 3. (IP version) Will this program compile and run without error? Why or why not?
- Question 3. (SP version) What is the output of the program (what is displayed on stdout)?
- Question 4. IP omitted
- Question 4. (SP version) The following warning is emitted when the code is compiled: "warning: ISO C++11 does not allow conversion from string literal to 'char *' [-Wwritable-strings] p="some text"; " In your own words, describe what the problem is and how it could be resolved.

These questions were strategically chosen to investigate specific aspects of student understanding and predicted or observed misconceptions. Specifically, it was hoped Question 1 (pointer declaration) would reveal important information about the maturity of understanding around pointers in C++, not just at a highly functional level, but also at a lower-level implementation.

Question 2 (assignment of pointer to string literal) would first show how students understand what a string literal is and how it is treated in memory and second, further test students' understanding of exactly what information is stored in a pointer.

Question 2 also required students to demonstrate a mature understanding of the context specific nature of operators. The correct interpretation of this code requires the student to understand that the data type of the operands have a critical impact on the operation associated with the operator.

Question 3 (sending data to standard output) further tests this notion. By outputting both the dereferenced value of the pointer in the first output statement, and using the pointer in the second output statement, the program examines the student's understanding of the stream insertion operator («).

Question 4 concerns the immutability of string literals in C++. Investigators anticipated that telling students about the compiler warning and asking them to interpret that warning would further highlight misconceptions around pointers and memory allocation. The answers to all these questions, it turns out, were quite enlightening.

3 Data and Analysis

3.1 Response Rubric and Assessment Scores

A rubric was created to rate responses on five criteria (numbered $C1 - C5$) aligned with the areas of investigation described in Section 2.1. The criteria are:

- $C1$: Exhibit advanced knowledge about pointer declaration and memory allocation for pointers
- $C2$: Exhibit advanced knowledge about string literals in memory and assigning char* variables
- $C3$: Indicate an appropriate level of knowledge concerning operators in different contexts
- $C4$: Demonstrate mastery of operator context knowledge by interpreting pointer dereferencing and indirection in sequence
- $C5$: Demonstrate knowledge concerning the immutability of string literals

For each of the five criteria, five levels of achievement were defined (with associated scores of 0-4). A score of four would represent a fully correct answer for each criteria. An abbreviated version of the full rubric is presented in Table 1.

The three investigators each scored all responses independently. The mean level of achievement for each criteria across each of the two data sets (IP and SP) was computed and is shown in Table 2. The data analysis, trends, and discussion of the data are presented next in Sections 3.2 and 3.3.

3.2 Sophomore-level Baseline for Student Understanding

The data accumulated from IP provided a baseline for student knowledge about the areas of investigation. While their prior coursework included pointers, memory allocation and memory management, achievement scores for IP highlighted opportunities for further growth.

For $C1$ these students scored a 0.94 average. While still novices at using pointers, IP respondents had all recently completed an intensive, extended period of instructional content requiring an appropriately advanced knowledge of pointers. Surprisingly, on average IP students were unable to give an answer to Question 1 more sophisticated than a naive definition of a pointer; mentioning little or nothing about how memory is handled when one is declared.

For $C2$ IP averaged 0.76. Of the 78 individual scores (26 responses × 3 reviewers) for $C2$, only two 3s were awarded, meaning no sufficient answers were given. Only ten 2s were awarded, and the remaining 66 scores were evenly

Table 1: Levels of Achievement by Criteria

	0	1	2	3	4
C1	No answer	declares a pointer	memory not mentioned	no memory content specified	memory allocated and initialized
C2	No answer	sets pointer to a string	describes copying C-string	assigns address to pointer but no memory allocation for string	describes memory allocation and address assignment correctly
C3	No answer	describes context dependent behavior of assign. op.	describes context dependent behavior of stream ins. op., all output wrong	describes context dependent behavior of stream ins. op., some output correct	describes correct output
C4	No answer	states won't compile	understands only dereference	understands only string indirection	response correct for both output statements
C5	No answer	assignment to pointer invalid	pointer wrong type	p should be array	assessment of warning correct

split between 0s and 1s. Recall that a score of 1 indicated answers that lacked specificity like, “sets pointer to a string”.

Scores for the third criteria were even lower. The $C3$ criteria measured students' understanding of operator overloading; that a specific operator's behavior is determined based on operands, not just the given symbol. The average score across IP students was a mere 0.36.

When asked to state if the sample program compiled, why or why not, the vast majority of IP students stated flatly that it would not compile. The average score for $C4$ for IP was a mere 1.18. A score of 1 for $C4$ was assigned to any student who said “no output was produced, or the program did not compile”, or similar. Of the 78 scores assessed for this question, only ten were 2's, (correctly identifying the output from pointer dereferencing) and only two

Table 2: Average Level of Achievement by Assessment Criteria

Data Set	C1	C2	C3	C4	C5
Intermediate Pop.(IP) (n=26)	0.94	0.76	0.36	1.18	N/A
Study Pop.(SP) (n=32)	1.70	2.01	2.21	2.27	1.46

were 3's (correctly asserting that the program prints out the string). No 4's were awarded.

Students in IP were not scored on $C5$. The authors' believed that the immutability of string literals would be outside the scope of reasonable expectations for IP students. The SP study results support that assertion.

It's clear IP students are far from possessing a mental model of pointers with the maturity necessary to understand higher-level notions. In and of itself, this is not particularly surprising. The study's goal was to determine how these concepts have matured for SP students. If SP students still possess common misconceptions, does it seem reasonable that they could be traced back to similar initial misunderstandings as those held by IP? Finally, if so, what implications for improved pedagogy does that suggest?

3.3 Senior-level Data and Notional Deficiencies

Since the SP students had all completed a CS2 course as a prerequisite, the authors expected to see a more mature understanding of pointers, data types, and memory allocation than that observed with IP students. Furthermore, other prior coursework, such as networking or software engineering should have motivated students to deepen their working knowledge of the examined topics.

The average SP score for $C1$ was 1.7. While slightly higher than the average observed with IP students, this clearly indicates a far from mature understanding of pointer declaration and memory allocation when a pointer is declared. Of the 96 individual scores (32 responses × 3 reviewers), only two 4s were awarded, and only three students received a 3 or higher from all reviewers.

SP students averaged 2.01 on $C2$. Of the 96 individual scores, only eleven 4s were awarded, with only 2 of the 32 students receiving a 4 from all reviewers, and only 6 students receiving an average score of 3 or higher.

For $C3$, around 48 percent (46) of the individual scores were 3s, and 31 percent (30) of the individual scores were 2s, indicating a slightly higher level, but clearly exhibiting persistent confusion about context-dependent behavior of the insertion and assignment operators. No individual scores of 4 were awarded, meaning that none of the SP students were able to demonstrate a complete understanding of the context-dependent behavior of the two operators.

The $C4$ scores for SP students demonstrated major gaps in students' knowledge pertaining to pointer dereferencing and C-string indirection. The average score for $C4$ was 2.27. Interestingly, 63 of the 96 scores indicated that students were able to correctly interpret C-string indirection but not the character dereference (17 of the 32 students received an overall score of 3). Only 4 of the 96 individual scores indicated that students interpreted character dereferencing correctly. No individual scores of 4 were awarded by any of the reviewers.

$C5$ scores indicate that SP students' understanding of a string literal's immutability was absent. The average score was 1.46 with 62 of the 96 individual scores being 1s or 0s, where students provided entirely incorrect answers or indicated that a string literal cannot be assigned to a pointer. Only 11 of the 96 scores were 3s, with only one student receiving an overall score of 3. No 4s were awarded, indicating that none of the responses mentioned string immutability or the need for the pointer to be constant as the solution to address the compiler's warning message.

4 Findings and Observations

Overall, the sampled responses indicated little growth in the examined topics. The authors propose, in the form of pedagogical recommendations, the following:

Suggestion 1: Emphasize the difference between declaring pointers and declaring memory for the pointers to reference.

As working CS educators, the authors acknowledge the effort put into teaching pointers. Early programmers are often taught the "mailbox" explanation of variables. Namely, that a variable can be conceptualized as a mailbox that can store a single value. Pointers confuse this metaphor. Unlike a mailbox, they reference where the value is stored, instead of directly storing the value. Moreover, assigning a single pointer to what, on the surface, appears to be multiple characters (in the form of a C-string-literal) further confuses the issue.

The authors find it telling that on an average exam in CS2, an acceptable answer to a question like, "What is a pointer?" would be to answer along the lines of, "A pointer is a variable that stores an address" (i.e. a mailbox). The results show this answer, while correct, obscures critical student misunderstandings. For example, the misconception that a pointer declaration automatically allocates the referenced variable, or the misconception that a pointer is a mailbox inside a mailbox. This unexpected revelation is one of the study's key findings.

Suggestion 2: Test students' understanding of what a data type is and what the implications are in various contexts.

Again, as CS educators the authors can attest to the fact that the formal definition of "data type" is dutifully drilled into CS students starting in CS0 and

CS1. However, results from this study suggest that memorizing the definition does not mean that students are correctly applying the definition in their code. For example, many students responded that displaying *p will output the string which suggests a deficient understanding about the data type concept.

Suggestion 3: Introduce function overloading in the context of the whole language, not just as a component of object-oriented programming.

The confusion for IP students in $C2$ appears to be similar to an issue which again shows up in the SP data. Students assume that the assignment operator can only have one meaning: take the value on the right-hand side and "assign" it to the left-hand side. Similarly, student responses indicated they believe that the stream insertion operator only has one meaning, display the value of the right-hand operand (open the mailbox and display the contents). This, in conjunction with the fact that these students mostly seem to have internalized the useful trope that, "pointers store addresses", demonstrate that it makes sense they would struggle to understand how a set of characters could be shoved into a "mailbox" for storing addresses.

The authors have observed a related issue with the division operator where $CS1$ students struggle with integer versus floating point division. Together, these observations suggest a more holistic approach is needed to illuminate the overloading concept.

Suggestion 4: Introduce C-strings and the immutability of string literals earlier.

The authors believe that understanding what must be done to translate source code into a working program is critical to moving from a novice to an expert. The second unexpected revelation in the data was where students responded with statements claiming that literals do not occupy memory or do not have an address and therefore could not be assigned to a pointer.

References

[1] Bruce Adcock et al. "Which Pointer Errors Do Students Make?" In: *Proceedings of the 38th SIGCSE Technical Symposium on Computer Science Education*. SIGCSE '07. Covington, Kentucky, USA: Association for Computing Machinery, 2007, pp. 9–13. ISBN: 1595933611. DOI: 10.1145/1227310.1227317. URL: https://doi.org/10.1145/1227310.1227317.

[2] Anthony Allevato, Stephen H. Edwards, and Manuel A. Pérez-Quiñones. "Dereferee: Exploring Pointer Mismanagement in Student Code". In: *Proceedings of the 40th ACM Technical Symposium on Computer Science Education*. SIGCSE '09. Chattanooga, TN, USA: Association for Computing Machinery, 2009, pp. 173–177. ISBN: 9781605581835. DOI: 10.1145/1508865.1508928. URL: https://doi.org/10.1145/1508865.1508928.

[3] Jonas Boustedt et al. “Threshold Concepts in Computer Science: Do They Exist and Are They Useful?” In: *SIGCSE Bull.* 39.1 (Mar. 2007), pp. 504–508. ISSN: 0097-8418. DOI: 10.1145/1227504.1227482. URL: https://doi.org/10.1145/1227504.1227482.

[4] Michelle Craig and Andrew Petersen. “Student Difficulties with Pointer Concepts in C”. In: *Proceedings of the Australasian Computer Science Week Multiconference.* ACSW ’16. Canberra, Australia: Association for Computing Machinery, 2016. ISBN: 9781450340427. DOI: 10.1145/2843043.2843348. URL: https://doi.org/10.1145/2843043.2843348.

[5] Reva Freedman. “Using an Operating Systems Class to Strengthen Students’ Knowledge of C++”. In: *Proceedings of the 51st ACM Technical Symposium on Computer Science Education.* SIGCSE ’20. Portland, OR, USA: Association for Computing Machinery, 2020, pp. 947–953. ISBN: 9781450367936. DOI: 10.1145/3328778.3366936. URL: https://doi.org/10.1145/3328778.3366936.

[6] Amruth N. Kumar and Rajendra K. Raj. “Computer Science Curricula 2023 (CS2023): Community Engagement by the ACM/IEEE-CS/AAAI Joint Task Force”. In: *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 2.* SIGCSE 2023. Toronto ON, Canada: Association for Computing Machinery, 2023, pp. 1212–1213. ISBN: 9781450394338. DOI: 10.1145/3545947.3569591. URL: https://doi.org/10.1145/3545947.3569591.

[7] Scott M. Pike, Bruce W. Weide, and Joseph E. Hollingsworth. “Checkmate: Cornering C++ Dynamic Memory Errors with Checked Pointers”. In: *Proceedings of the Thirty-First SIGCSE Technical Symposium on Computer Science Education.* SIGCSE ’00. Austin, Texas, USA: Association for Computing Machinery, 2000, pp. 352–356. ISBN: 1581132131. DOI: 10.1145/330908.331884. URL: https://doi.org/10.1145/330908.331884.

[8] Yizhou Qian and James Lehman. “Students’ Misconceptions and Other Difficulties in Introductory Programming: A Literature Review”. In: *ACM Trans. Comput. Educ.* 18.1 (Oct. 2017). DOI: 10.1145/3077618. URL: https://doi.org/10.1145/3077618.

The Digital Twins Incident Response To Improve the Security of Power System Critical Infrastructure[*]

Abimbola Akerele[1], William Leppert[2]
Shionta Somerville[3], Guy-Alain Amoussou[3]
[1]Department of Computer Engineering
Prince George's Community College, Largo, MD 20774
bimzykay7@gmail.com
[2]Department of Computer Science
Virginia Western Community College, Roanoke, VA 24015
wtl224@email.vccs.edu
[3]Department of Computer Science
Bowie State University, Bowie, MD 20715
{ssomerville, aamoussou}@bowiestate.edu

Abstract

Digitalizing real-world assets has numerous benefits, including the algorithmic analysis of physical objects. Digital twin (DT) technology has evolved to further contribute with continuous real-time data valuable flow to the analysis. With this analysis comes a greater understanding of a problem, which, in turn, provides a more comprehensive solution. We will examine the definition of critical infrastructure, to understand better how DT technology can be applied in a large-scale physical environment. We then turn our focus to the components within the energy sector of critical infrastructure, including various Industrial Control Systems (ICS) components with a focus on sensor networks and Supervisory

Control and Data Acquisition (SCADA) systems. With this understanding, we examine research pertinent to the key features of a DT and how that may be applied to the sensor and ICS networks. Field sensors within ICS are susceptible to semantic attacks leading to disruption of horizontal and lateral operations and systems. With this possibility, incident response algorithms can be implemented in a digital space of an energy grid to maintain security and operability during adverse conditions. This review intends to outline these core components for additional lab testing and research, analyzing the effectiveness of digital twins in securing operations of Industrial Control Systems in the energy sector.

1 Introduction

The appreciation of digital technologies designed to monitor and control crucial functions of critical infrastructure (CI), comes the increased necessity to design systems engineered to ensure those functions. When examining the present state of cyber security, various tools and methods can be applied across sectors to support the overall operations of critical interconnected infrastructures. As networks of industrial control systems (ICS) continue to grow, a technology that scales with this growth while maintaining system-wide visibility is necessary. Since the beginning of NASA's space exploration, pairing technology has been implemented to digitize physical objects for computational methods[9]. This technology was the predecessor to what is now referred to as a digital twin (DT). DTs have seen successful applications in automotive and aviation manufacturing and are expected to continue to expand across industries. The global market value of DTs was estimated to be $5.04billionin2020andisexpectedtoreach86.09$ billion by 2028, with the manufacturing and energy industries leading in DT deployment at 34 percent and 18 percent, respectively[10]. Computational methods applied within a modern digital environment, on which we will focus our research, are found in incident response algorithms. Within a DT environment, incident response algorithms can be designed to establish safeguards and remedies against security threats. One such threat that will be the focus of our research will be semantic attacks[3]. Considering this, we aim to provide a functional analysis of the implementation of incident response algorithms within a DT environment. This environment's objective is to improve the visibility and security of ICS in the energy sector when faced with a semantic attack.

2 Digital Twin Incidents Response Methodology

Defining the crucial phases of research process by categorizing the phases into the formative phase, the experimental phase, and final phase. The formative phase describes the foundational background of our topic. The first phase was gathering qualitative data that formed the fundamental base of the essential component of research. The experimental phase explains the hands-on process stage of the process. This involves digitally and physically practicalizing the process of deploying a digital twin that is then updated with computer-generated data or data from physical devices. Finally, phase 3 involves deploying the power system modeled in Digital Twin Definition Language (DTDL) to Azure DT. That is then updated through the simulated operations controlling frequency, voltage, and phase difference between two nodes in the modeled power system.

2.1 Formative Phase

This foundational, fundamental, and formative phase exposed us to the basic knowledge and critical components relevant to our research by gathering quantitative data on our topic from other scholars. This includes the meaning, function, and importance of digital twins (DT), the 16 critical infrastructure systems of the United States, digital twins in transport systems, and traffic control systems. At this point, it was crucially necessary to set a distinguishable boundary between digital twins and simulation. A digital twin involves a two-way communication of real-time data between the physical and digital twin, which goes only one way. Understanding this difference helped direct our focus for the physical experimental phase with DT software and raspberry pi we would be facing subsequently. We further explored the importance of Industrial Control Systems (ICS) through SCADA systems in critical infrastructure processes and the need to protect these systems from cyber-attacks to prevent catastrophic national meltdowns[11]. While gathering this information and discovered that the earlier research mainly focused on protection and detection. Only some research was gathered on a countermeasure or response in case of an incident [7]. The analysis of incident response architecture composed of the Software Defined Network (SDN) and the Network Function Visualization (NFV) to foster a defensive and secured system. This phase of our research served as a solid bedrock for adding theoretical and practical phases essential to generating results in our research.

2.2 Experiment Phase

The next phase of our research involved applying the knowledge gathered in the formative phase. During this phase, we explored the available digital twin's software and established connectivity with its physical twins. It is important to identify the challenges faced during this phase.

- Digital Twin Software: AWS Digital Twins Initially, we experimented with the Amazon AWS digital twin software to get a hands-on experience using provided tutorial documentation available on the internet. We determined AWS DT to not be the most cost-effective approach. In addition, we discovered that not only was the website difficult for new users to navigate but there were also less resources available for new users to understand the use of AWS DT. For this reason, we decided to switch our focus to the Microsoft Azure Digital Twin.
- Microsoft Azure Digital Twins: Opened accessible accounts, opted for the free subscription to save cost, and opened new instances and storage accounts. We noticed simple and understandable walkthrough documentation readily available for new users of Azure DT to get a quick, easy, and straightforward overview of the working process of the technology. We had access to simple models and components of 6 robot hands, room and floor, and a digital realistic 3D Scene of a package distribution industry. A step-by-step guide on deploying each piece gave us a hands-on introduction. Due to the time constraint on our research, we had to redirect our focus to another crucial part of our experimental phase. We began to implement a physical twin with connection capabilities to the Azure DT.
- Physical Model: Raspberry Pi: Constructed a small-scale physical representation of a component in the power grid by first verifying connection capabilities to Azure DT with a DHT-11 temperature and humidity sensor. This sensor was connected to a breadboard to a Raspberry Pi B+ with a 40-pin GPIO cable. Through available resources, we initially attempted connectivity from the physical representation to Azure IoT Hub and from the IoT Hub to Azure DT with Azure Function App. In our configuration, the IoT Hub registered updates from the sensor but did not register Function App events and therefore did not update Azure DT. This approach was discarded, and we sought a direct connection to Azure DT through the python library azure-digital twins-core. This satisfied the requirement to verify connectivity, and we constructed the representation of the selected power grid component, a Hall effect sensor, and a power relay. This component was designed to represent current measurement from the Hall effect sensor with subsequent relay operations. This relay operation was determined by activating the Hall effect sensor at a particular magnetic field level. As in, at a particular level of current, a power relay would activate to open or close a circuit. A voltmeter was added for digital verification of relay operations. This construction was delicate and thus difficult to change. The Hall effect sensor and relay were operational, but the voltmeter was not operating as expected. While

troubleshooting the voltmeter operation, the device was short-circuited, leaving the breadboard and the GPIO connection pins inoperable. Due to this challenge, we have since pivoted to construct a digital representation of the power grid, focusing on the simulation of voltage, frequency, and phase difference regulatory operations.

- Phase 3: Within a digital representation, there is greater capacity to scale and apply to actual distributed power systems. To begin this process, it was understood how larger components within the transmission and distribution sectors of the power grid would deliver power to the customer. From that then developed the graphical representation, from distribution to a step-up transformer down high voltage transmission lines to a substation and step-down transformer. The distribution sector was often organized into 3-phases, with phase 3 being the initial distribution from the substation transformer, phase 2 being connecting lines in the distribution, and phase 1 lines being the final endpoint distribution. For simplicity's sake, the current digital model primarily represents voltage regulation at one phase. Semantic attacks, or data falsification, of voltage measurement can be achieved by data injection, alteration, blocking, detection and/or modification [1]. As cited in [1], data falsification attacks on voltage can be additive, deductive, or a combination. The normal parameter regulation behavior in our current model is strictly programmatically defined, issuing increasing, decreasing, or stabilization commands to current parameter values through a generic parameter controller. Stabilization is performed by finding the difference of the current value and the average of the most recent five values. If this difference is greater than a predefined quantity, an increasing or decreasing operation is performed to return the current value closer to the historic average. This is without issuing actions taken by specific regulator types used throughout the modern physical distribution grid. After gathering additional information on semantic attacks on voltage, frequency and phase measurements and simulating parameter control coded in python, the simulated control logic shown below in Figure 5 was then exploited to display the implications of data falsification[3]. The occurrence of the alteration of data will be the focus of our detection algorithm. Then performed additive and subtractive testing to examine behavior. It is from this we intend to continue developing our understanding of grid visibility within a digital twin environment aimed at the implementation of response algorithms designed to provide safeguards and remedies to security events and incidents[8]. An additional feature to examine within this model would be SCADA operations run at the previously mentioned substation to work in tandem with digital twin data and operation functionality[2]. With historical data from multiple sensor devices integrated with a SCADA system along with additional metrics such as meteorological forecast. This data can potentially be used to analyze the probability of behavior within a system of measurements. Within the digital twin structure, anomaly detection and classification through machine learning algorithms are the focus of [4] and is the primary reason Phasor Measurement Units (PMUs) have become the focused field device of this research. An analysis such as this can be designed around the recorded nominal behavior of a power system in the DT environment. Then, we begin to classify anomalous events appropriately for further evaluation and produce actionable identifiers to commit within the response algorithm. Phasor Data Concentrators (PDCs) process data sent by PMUs and have been shown to be susceptible to false data injection as displayed in [4]. The combination of the attack methodology on voltage regulation and the time synchronized values of voltage magnitude, phase angle, and frequency measured in PMUs were the basis of our control simulation and attack simulation. The following will describe the setup.

3 Digital Twin System Result

This section showcases and displays the impact of an attack on three measurement variables used in power system operations: frequency, voltage, and phase difference. A basic power system was modeled in Azure Digital Twin and updated with azure-digital twins-core from a group-constructed simulation of system operations written in Python. These results, along with our extensive research, have led us to the future research proposal detailed in the final section.

Figure 1 represent the basic components of the power grid model: A power generation plant, step up transformer, substation, step down transformer, industrial breaker box, voltage regulators (for step down transformer and the breaker box), and the endpoints of the industrial office and industrial machinery. Each of these components have been implemented into the Azure DT environment, where they are active as twinned entities. In Figure 3, the data handled by each entity can be shown. Within the simulation, the variables described are verified within a defined tolerance attributable to real-world grid operations. Figure 3 displays the current function implementation, integrating our simulation and digital twin model.

The simulation is made up of parameter values for the variables that have been manually generated to test tolerance conditions. If they are not, the associated processes are performed; three flags and one action are outputs of these processes along with the controlled value. A high-priority flag is associated

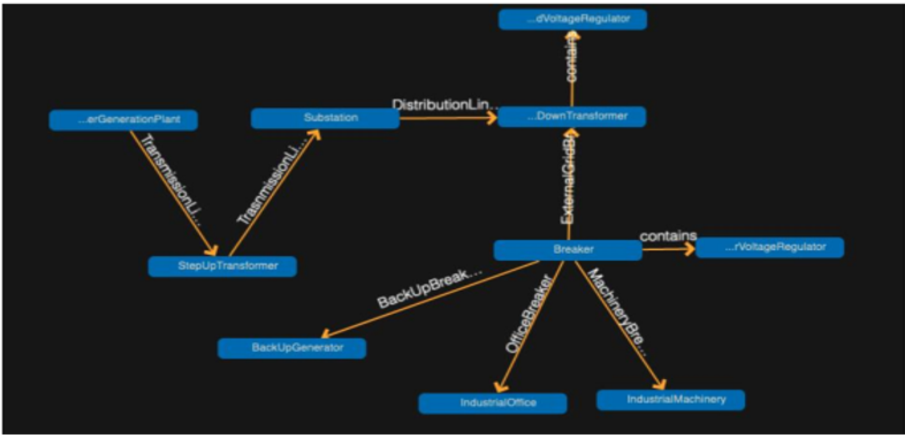


Figure 1: Azure Digital Twin Model Relationship.

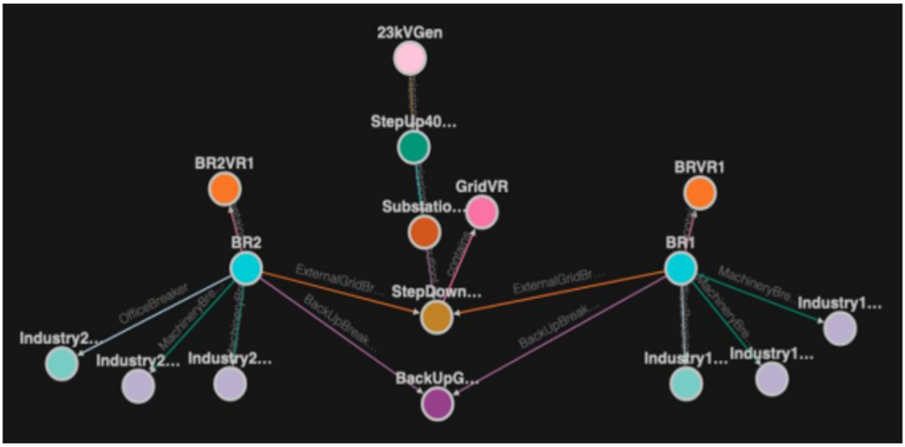


Figure 2: Azure Digital Twin Graph of Sub-grid of Two nearby industrial sites.

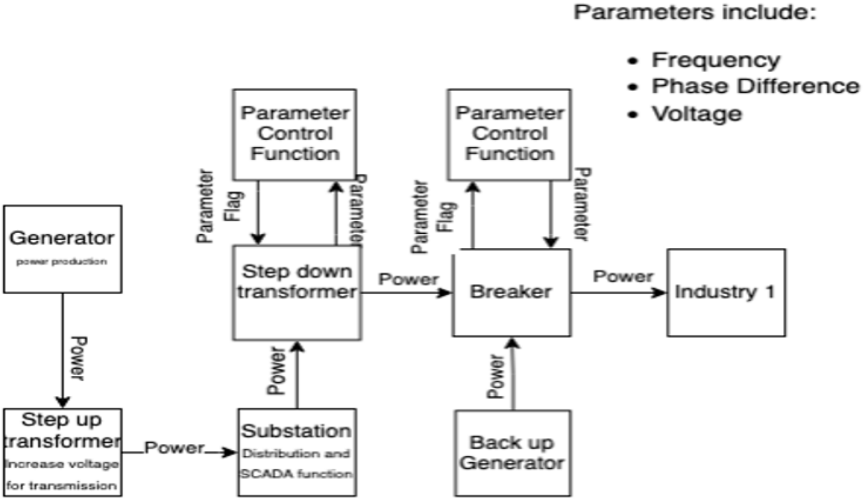


Figure 3: Integrated Functional Block Diagram.

with the action of disconnection from the power source due to the values exceeding the tolerance in a way that has been deemed uncorrected. Moderate priority flags are associated with values slightly exceeding the tolerance as well as if the values fluctuate dramatically. A generic graphical representation of this control functionality is presented in Figure 4.

The controlled values generated through a simulated phasor measurement unit (PMU)-based control system determining control operations of frequency, voltage, and phase difference are shown in figure 5.

The values were manually generated and assumed to be raw measurement readings. Based on the assumption, it was presumed that the attacker gained access to the system and manipulated the raw measurement readings at will. The manipulation matches the feedback within the simulated frequency, voltage, and phase difference control mechanisms, adversely affecting the intended output.

3.1 Digital Twin Incident System Design

The designed structure displays the impact data falsification would have on controller operations. Each variable has been treated as independent, as in, change to one would have no effect on the other. As displayed in Figure 6, the results from injecting a measurement reading at the maximum tolerance will cause the controller to steadily decrease the actual output until it has fallen

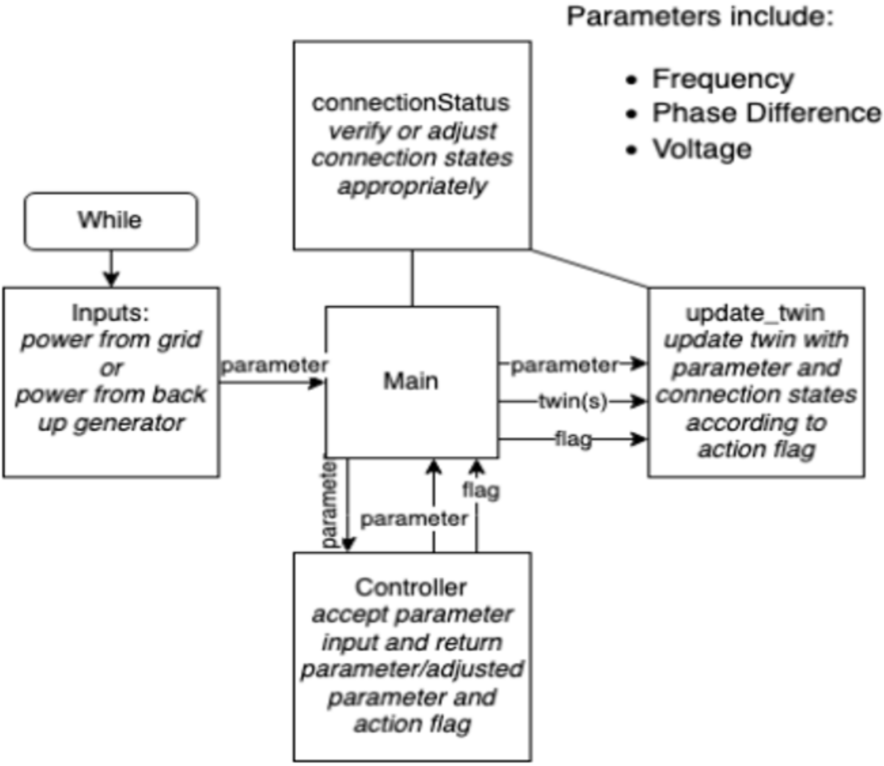

Parameters include:
• Frequency
• Phase Difference
• Voltage
connectionStatus
verify or adjust connection states appropriately
While
Inputs:
power from grid or power from back up generator
parameter
Main
parameter
twin(s)
flag
update_twin
update twin with parameter and connection states according to action flag
parameter
parameter
flag
Controller
accept parameter input and return parameter/adjusted parameter and action flag

Figure 4: Simulation Functional Block Diagram.

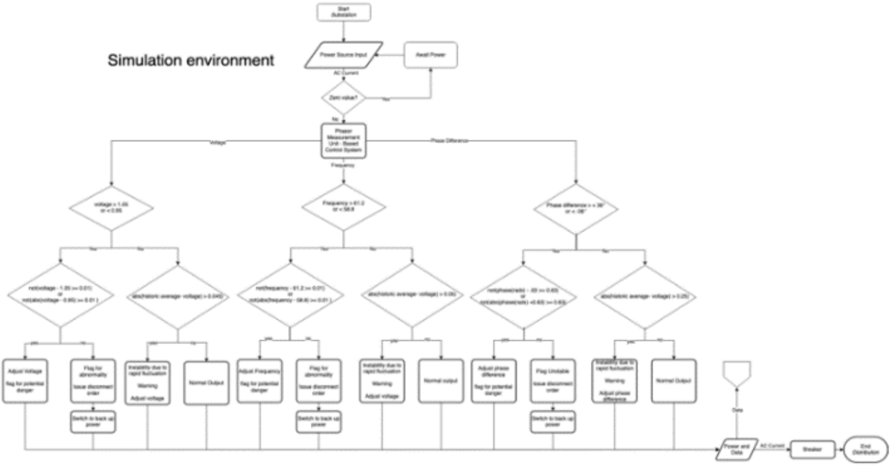


Figure 5: Simulation Logic Flowchart.

below the minimum tolerance. In Figure 7, a deductive attack is represented, causing the voltage to exceed the maximum.

In the case of undervoltage or overvoltage, the effects can either, respectively, cause overloading of components as amperage increases or cause components to lose insulation and short circuit, as mentioned in [6]. Both phase difference and frequency are displayed concurrently in the next section.

The impact within the simulation can cause both phase difference and frequency to fall below or exceed their minimum and maximum tolerances. While these variables were treated independently, they are related.

For simplicity's sake, we will consider a single-phase alternating current line as opposed to the traditional three-phase lines offset by 120°. Within this single-phase line, the phase difference, a product of changes in frequency, between two nodes indicates consumption or generation of power. If the downstream node lags the upstream node, then more power is consumed than produced. If the upstream node lags the downstream node, more power is produced than consumed.

Frequency and phase difference are indicators of grid stress and stability. To keep the grid in balance, these variables are controlled through various operations, largely, we have found, by power injection techniques. The Figures (8 – 9) display the impact of these control operations when under cyber-attack through data falsification. The resulting impact could lead to partial or complete blackout as mentioned in [5] and [7].

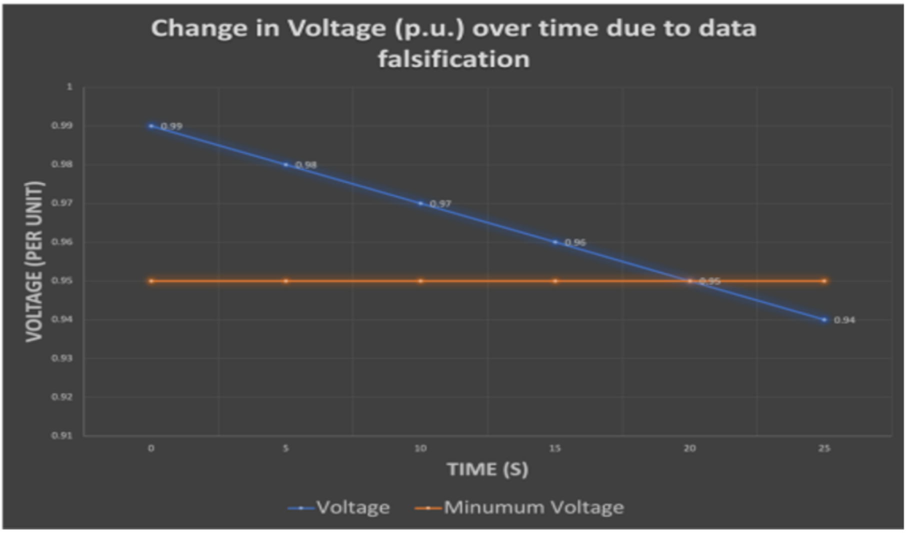


Figure 6: Addictive Attack Results Voltage

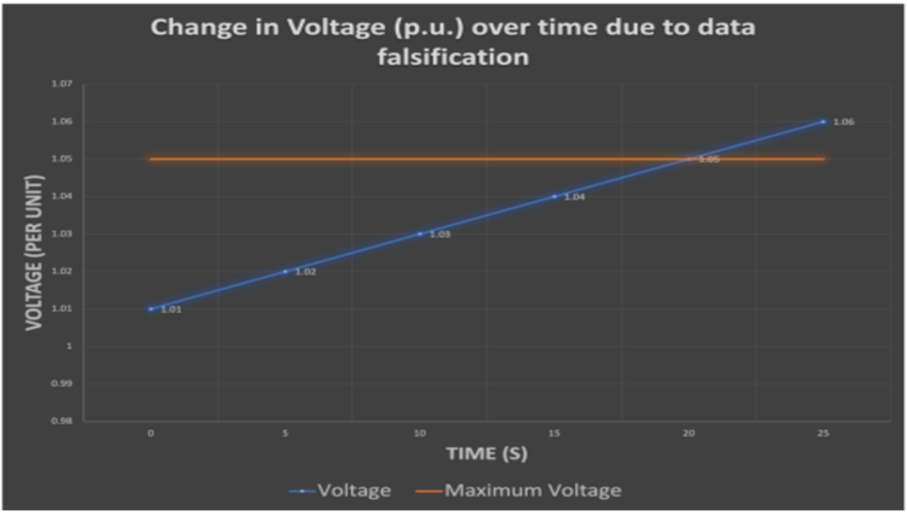


Figure 7: Declarative Attack Results - Voltage

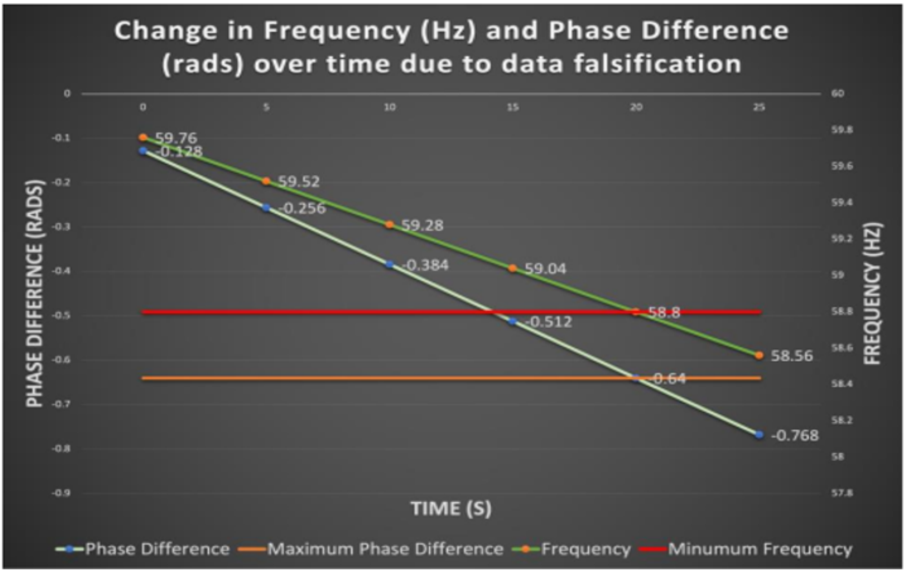


Figure 8: Addictive Attack - Frequency and Phase Difference

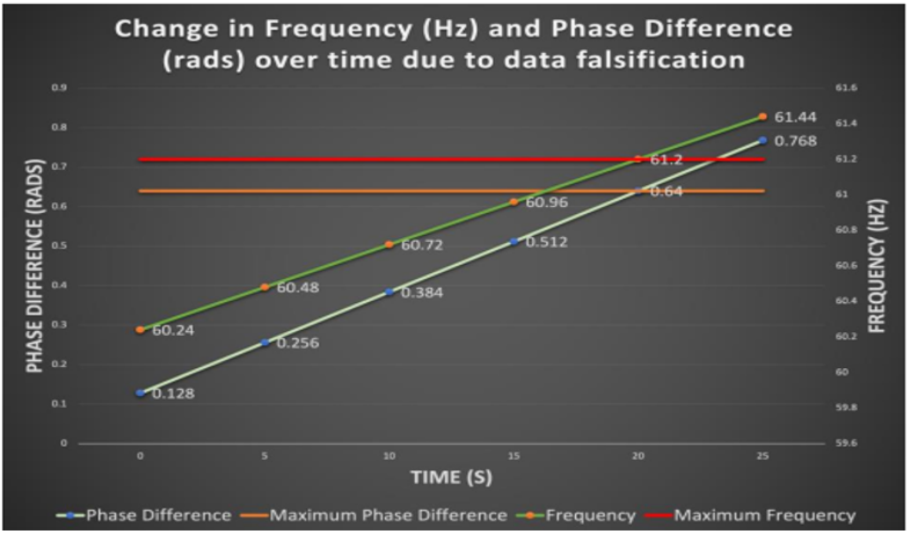


Figure 9: Deductive Attack -Frequency and Phase Difference

3.2 System Design Challenge

The Amazon Digital Twin Maker seemed to be difficult for new users, and switching to the Microsoft Azure Digital Twin software proffered a solution to the digital twin challenge for our team as first-time technology users.

The challenge eventually caused me to switch focus to the digital representation due to several limitations of the physical components. After successfully testing and validating connectivity in the DHT-11 humidity and temperature sensor connected to a Raspberry Pi using the python library azure-digital twins-core, we tried to connect the hall effect sensor and a power relay to represent current measurement from the hall effect sensor. A voltmeter was added for digital verification of the operations. The voltmeter was not operating as expected, and while troubleshooting, the device was short-circuited, leaving the breadboard and GPIO pins dysfunctional. Also, SSH connectivity from the Raspberry Pi with the Windows OS is another challenge that led us to explore the digital representation of the project.

Worked on the physical components using the hall sensor and relay modules, we designed a graphical representation of the model, functions, and connectivity. However, this representation was discarded because the diagram was limited and not clear to understand.

4 Discussion and Conclusion

The focal point was US critical infrastructure, and from there, we narrowed our research to the energy sector, its importance, and its working principles. Since our research emphasizes security through digital twins, we placed emphasis on papers that dealt with monitoring, detecting, and mitigating attacks on ICS smart grid systems, primarily through digital twins. The Microsoft digital twin software by exploring simple, quick guides. This allowed us to create our models, components, and characteristics based on research for a power grid. To test a real-time data streaming process in the digital twin, we connected a physical device, the DHT-11 temperature and humidity sensor, which derived real-time data from the environment and updated the twin. The challenge faced led to a pivot to digital representation. Further, the research was deepened on the power grid connection and complexities from a general and specific point of view, considering the vulnerable components to an attack. And it was noticed that the vulnerabilities of ICS systems on power grids are increasing as grids expand and become smart with IoT integration. Moving on to the digital twin, let's develop models and twins of these found components, showing their relationships to one another and variables that are often affected in the case of an anomaly. The variables include frequency, phase difference, and voltage.

Monitoring the range in these variables and noting a slight or massive shift will help determine an anomaly. To precisely simulate a real-world attack, there was the need to research incidents, attack vectors, variables, and components affected as well as the impact of the attack on the system, and this clearly defined these scenarios and use cases to assist in identifying the best use case to simulate an attack on the developed system. It was also discovered that stress on the grid ultimately causes the grid to break down.

5 Acknowledgement

The National Science Foundation, Award 1757924, supported this research.

References

[1] Narayan Bhusal, Mukesh Gautam, and Mohammed Benidris. "Detection of cyber attacks on voltage regulation in distribution systems using machine learning". In: *IEEE Access* 9 (2021), pp. 40402–40416.

[2] Peter Eden et al. "A forensic taxonomy of SCADA systems and approach to incident response". In: *3rd International Symposium for ICS & SCADA Cyber Security Research 2015 (ICS-CSR 2015) 3*. 2015, pp. 42–51.

[3] Eleni-Maria Kalogeraki, Spyridon Papastergiou, and Themis Panayiotopoulos. "An attack simulation and evidence chains generation model for critical information infrastructures". In: *Electronics* 11.3 (2022), p. 404.

[4] Ehdieh Khaledian et al. "Real-time synchrophasor data anomaly detection and classification using isolation forest, kmeans, and loop". In: *IEEE Transactions on Smart Grid* 12.3 (2020), pp. 2378–2388.

[5] Tim Krause et al. "Cybersecurity in power grids: Challenges and opportunities". In: *Sensors* 21.18 (2021), p. 6225.

[6] P Rama Mohan et al. "A Novel Over Voltage and Under Voltage Protecting Systemfor Industrial and Domestic Applications". In: *International Journal of Innovative Science and Research Technology* 5.10 (2020), pp. 885–889.

[7] Vetrivel Subramaniam Rajkumar et al. "Cyber attacks on power system automation and protection and impact analysis". In: *2020 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe)*. IEEE. 2020, pp. 247–254.

[8] Andrey Rudskoy, Igor Ilin, and Andrey Prokhorov. "Digital twins in the intelligent transport systems". In: *Transportation Research Procedia* 54 (2021), pp. 927–935.

[9] Arindam Sarkar, Mohammad Zubair Khan, and Abdulfattah Noorwali. "Chaos-guided neural key coordination for improving security of critical energy infrastructures". In: *Complex & Intelligent Systems* 7 (2021), pp. 2907–2922.

[10] Maulshree Singh et al. "Applications of digital twin across industries: A review". In: *Applied Sciences* 12.11 (2022), p. 5727.

[11] Keith Stouffer, Joe Falco, Karen Scarfone, et al. "Guide to industrial control systems (ICS) security". In: *NIST special publication* 800.82 (2011), pp. 16–16.

Incorporating Scrum Concepts in the Capstone Course[*]

Jeffrey C. Jackson
Department of Mathematics and Computer Science
Duquesne University
Pittsburgh, PA 15282
jacksonj@duq.edu

Abstract

Although many undergraduate computer science programs in the U.S. incorporate a capstone project course, surprisingly little seems to have been written regarding experiences with various approaches to leading such courses. This paper addresses this paucity in a small way by describing the structure of a capstone course that the author has delivered twice with small variations. The course incorporates several elements of the popular Scrum agile development methodology. Both course offerings received high student-survey scores and were much more satisfying to the instructor than many earlier capstone-project experiences had been.

1 Introduction

It seems that many, if not most, undergraduate computer science programs in the United States require some form of capstone experience. The author's institution in recent years has moved from an independent-study style experience to a capstone course. In reviewing the literature while preparing for this move, a few papers came to light, including [4], [2], and [3]. These papers proved helpful; for instance, these thoughts from [3] influenced part of the course design, as will be evident below:

> A good project should have at least three separate phases or stages or partitions in the supervisor's mind when the project is designed or formulated. Firstly there should be some known and quantifiable task or part of the work that an average student can work on and be expected to complete. Secondly there might be the expected stage or partition of the project that a "good" student ought to be able to tackle, make sense of, and do a "good" job of. Finally there should be some aspect of the project that a "star" student should be able to "shine" in.

A review of U.S. final-year projects in [2] suggested that individual projects are typical. This meshes well with the observation of [3] that "Unsurprisingly, we have found that the more a student feels 'ownership' of their project, the more effort they are likely to put into the work and it will quite possibly be of higher quality." [4] addresses the purpose of a final-year project, starting with this goal: "It should prepare students for the working life, making them familiar with the work place by practicing their skills on real-world problems." Although [4] goes on to list additional possible goals for a capstone course, this first one seemed most relevant to the author's program and students. This goal is related to the "enculturation" objective for final-year project courses proposed by [1].

In light of these and other thoughts, key objectives for the new course were:

- Allow students freedom to choose their own topic within constraints that would facilitate meaningful faculty review.
- Provide somewhat objective assessment measures to guide students in their time allocation and provide them with a sense of the effort needed to secure a desired grade.
- Make good use of in-class meetings.
- Incorporate mechanisms to encourage students to produce good-quality artifacts and to facilitate reviews of the quality of the work.
- Introduce students to current software industry "culture," with an emphasis on norms regarding development processes, testing, code documentation, presentations, and reviews.

While the literature proved helpful in *defining* course objectives, it seemed to provide less guidance regarding how to *achieve* these objectives. This paper describes an approach based on the Scrum software development process that has worked well for the author. First, key elements of Scrum will be covered briefly. Next, a capstone course that uses Scrum concepts in addressing the objectives above will be presented. The paper concludes by reviewing student feedback on the course.

2 Scrum Elements

In order to address both the objectives of facilitating review and introducing students to software industry culture, the course incorporated several elements of the popular Scrum agile development process. Scrum originated as a general business practice [6] and was subsequently adapted for use in software projects; see [5] for an example of early adaptation work. Although typically viewed as a method of organizing teams of software developers, a number of Scrum concepts can be used fruitfully in the context of individual student projects. The concepts, as defined for purposes of the capstone course, included the following:

- **User story**: A Scrum user story is typically a one-sentence description of a feature of value to the end user of a software system. For instance, in the context of an automated homework system, a user story might be "As a homework-set author, I want to be able to write fill-in-the-blank questions." As this illustrates, a user story states the type of user and a feature of interest to this user. User stories are a primary outcome of the analysis activity in Scrum and are also used in managing Scrum projects.

- **User Acceptance Test**: A user story is considered a starting point in the analysis process. A more detailed view of a user story's requirements is provided by writing user acceptance tests. Each test specifies one or more actions that a user will take and the expected system responses. For instance, a user acceptance test for the above example user story might direct the user to log in as a specific instructor, guide the user through creating a specific fill-in question, and explain how to verify that the question has been successfully created. Descriptions of the system's responses throughout this process would be provided. A user acceptance test passes if all of the expected responses are observed. Typically, a suite of user acceptance tests will be written for a user story, with at least one test covering normal processing and others covering atypical situations such as the response to erroneous inputs.

- **Task**: When a user story is selected for implementation, it is decomposed into a collection of tasks. For the fill-in question user story, tasks might include, among others, "Design user interface," "Add fill-in question type to database," and "Write user acceptance test suite." Tasks should be small enough to be estimable with a reasonable degree of certainty, and a time estimate should be included with each task.

- **Definition of Done**: A definition of done for a user story specifies the conditions that must be met for the story to be considered complete. A typical definition of done specifies not only that the code exhibit the functionality required by the story but also that it meet specified quality standards, such as that tests cover at least 90% of the lines of code.
- **Sprint**: A Scrum sprint is an atomic slice of the project. Each sprint has the goal of completing—meeting all of the definitions of done for—a set of user stories by a set date. The set of user stories is chosen by the development team and is frozen for the duration of the sprint.

Although not an element of Scrum itself, another widely adopted "cultural" norm in modern Scrum-based software development is **continuous integration** moderated by **software reviews**. The basic idea is that team members should be incorporating their work into a shared code base whenever they have something that is ready for use by the rest of the team. This approach avoids the "big bang" that can occur when each team member creates their own components in isolation and only near the end of a development phase attempts to integrate these components with those of other team members. However, when employing continuous integration, it is also important to avoid "breaking the build," that is, incorporating into the shared repository work that is so flawed as to prevent the rest of the system from functioning properly. Hence, it has become standard industry practice that when a developer wants to integrate their work into a shared repository, it must first be reviewed and approved by at least one other team member. Software reviews are also used to verify that quality standards for code, tests, and documentation are being followed.

3 Capstone Course Description

The course that will be described has been offered twice, in the springs of 2021 and 2023. The course offerings were in many respects very similar, so this section will for the most part describe them as a single course. The ways in which reviews were performed were very different, and these differences will be explained. For other, smaller, differences, the second course's approach is described.

3.1 Project Selection Requirements

The course involved students working individually on projects of their choosing, subject to instructor approval. As noted earlier, individual projects have the benefits of promoting a sense of "ownership" as noted by [3].

However, although students had broad latitude in choosing their project, there were a number of restrictions designed to facilitate the course goals stated previously. For instance, it was required that the professor and classmates be able to compile and run the project code on lab machines or the equivalent. This requirement was intended to partially address the concern about superficial supervision potentially leading to poor-quality elements of projects. Admittedly, this requirement also likely steered some students away from projects that they might have found more interesting than the ones they proposed. Still, students seemed pleased to know that their work was being exercised by at least their instructor and potentially their classmates. And so far, no student has complained about this limitation.

Another requirement was that the project be amenable to automated testing. This requirement was similarly motivated by a desire to see students focus on the quality as well as the functionality of their software. That said, user interface (UI) testing tends to be trickier and, in the author's experience, more brittle than functional testing. Therefore, all students were asked to try to implement UI testing but also allowed to discontinue these efforts if the further time investment seemed likely to exceed the quality/learning benefit to be gained. Interestingly, the best students seemed to take it as a challenge to achieve high coverage of their entire code base, UI included. This gave a nice balance between providing a rich testing experience to some students while not miring other students in an endless morass of testing complexities.

Yet another requirement was that students use an approved programming language. The use of languages that were reasonably familiar, or at least intelligible, to all students facilitated their ability to review one another's code. Also, the instructor was in a better position to potentially guide students in resolving technical difficulties that they might encounter during their project development.

3.2 Project Initiation

Scrum concepts played a particularly important role in the early stages of the projects. Once students had selected a tentative project within the given constraints, they were asked to do several things. First, they wrote a high-level description of the project. They then wrote user stories describing the features that they hoped to implement over the course of the semester. Next, they partitioned the user stories among three milestones; the target date for the first milestone was set at about six weeks into the semester, the second at about ten weeks, and the third at semester's end. They then decomposed their Milestone 1 user stories into tasks and estimated the number of hours each task would require to complete.

To help in the decomposition of user stories into tasks and estimation of time requirements, a user-story definition of done checklist was provided to the students. This checklist indicated that students would be required to write user acceptance tests and have them pass. They would also be expected to write code that met quality criteria such as being free of static-analysis warnings, following object-oriented design principles as appropriate, and employing good code style with respect to indentation, variable names, and the like. Other requirements included writing automated tests with high coverage of lines and writing xdoc comments (Javadoc, pdoc, etc.) of all classes and methods.

Finally, based on their task estimates, students were asked to adjust their Milestone 1 user stories as needed so that the estimated hours fell within a certain range; the instructor recommended 20–28 hours for this course. Overall, the students were required per the course catalog description to spend at least 100 hours of effort on the project outside of class. Thus, 33 hours might be thought to be an appropriate target for Milestone 1. However, in the author's experience, almost all students underestimate the time required to produce software that will be accepted as done. It therefore seems best to assume that students will spend more time on this milestone than they expect.

Each student documented their work in a repository that the instructor created for that student at the popular cloud-based repository store, GitHub. In particular, the instructor retained ownership of each repository, protected the main branch to facilitate code reviews as described later, and defined an initial development branch for the student to work on. GitHub provides some basic project management tools that were used to document student work. Specifically, students wrote their project description in the development branch README, the user stories were represented as GitHub projects, tasks for the Milestone 1 user stories became GitHub issues that were linked to a GitHub Milestone 1, and the GitHub Milestone 1 description was edited to include a list of its projects/user stories.

After reviewing and recommending changes to student work—for instance, many students had difficulty writing good user stories on their first attempt—the instructor met with each student to discuss their Milestone 1 proposal. Either as part of this meeting, or sometime afterward if the instructor felt that changes were needed, Milestone 1 was formally approved for each student. Although the term "milestone" was used for consistency with GitHub, in Scrum terms the Milestone 1 for each student more-or-less defined their first sprint. A key difference was that the "sprint" did not end at the milestone target date. Instead, if the user stories in a student's Milestone 1 were not done by the target date, the student continued work on those user stories until they were completed. The target date thus served as a goal and, if passed, a progress marker, not as a deadline.

Before development started, students were directed to information about software architectural styles, instructed to select a style, and asked to document a high-level design for their system in terms of that architecture. This was intended to help students clarify their thoughts about how to begin their coding, give the students a conceptual organizational framework for their code base, and assist reviewers in navigating the code. It took the place of some aspects of what in Scrum would be a team's planning meeting at the beginning of a sprint.

3.3 Project Supervision

Students were encouraged to work outside of class on their project for seven to eight hours per week (this number being based on the 100 hour semester goal). At the end of each week and along the lines of continuous integration, students were expected to submit a reasonably clean version of their work at GitHub for review.

The instructor or an assistant would then check that various criteria of the definition of done were met. The reviews often led to the instructor/assistant requiring revisions. After possibly multiple revisions, the work would ultimately be approved for inclusion in the main branch of the student's GitHub repository, signaling that the work was of acceptable quality per the definition of done.

Hopefully, but not always, this merging into main would occur before the student's next week of work was submitted for review. However, if not, the GitHub review-request mechanism (*pull requests*) separated each review request and its associated student/instructor conversation from any other review requests that might be active at any given time.

A very helpful feature of the GitHub pull request facility is that it displays side-by-side the differences between the code as it existed prior to the pull request and as it exists after the modifications included in the pull request. This feature greatly facilitates a reviewer's ability to focus on only the portions of the code that need the reviewer's attention. When a student felt that they had completed the first milestone, they included this information in their pull request. In this case, not only the quality of the work was evaluated but also how completely it implemented the milestone's user stories.

A key difference between the first and second course offerings had to do with the form of the reviews. In the first offering, the instructor met with each student one-on-one every week. During this meeting, the student would walk the instructor through the changes for that week and the instructor would provide immediate feedback both orally and by writing review comments in GitHub. While this approach had some advantages, there were also problems. A major issue for some projects is that they often became help sessions more than reviews. Although this arguably had instructional benefit for the students, the review component was given limited time and therefore tended toward superficiality. In the second offering, the other extreme was attempted: Reviews were done entirely asynchronously via GitHub. A problem with this approach was that it seemed very inefficient at times—a question would come up that would be simple to answer interactively but that would break the flow of the review when posed asynchronously. This sometimes led to the reviewer taking time to attempt to answer the question on their own rather than waiting for a response. At least one student comment also suggested that students would have appreciated more direct interaction. In short, it seems in retrospect that a blend of short interactive sessions followed by deeper dives into the updated project would perhaps be a better approach to reviews.

Once a student completed Milestone 1, they considered their remaining user stories (the "product backlog," in Scrum terminology). Although they had tentatively selected a set of user stories for Milestone 2 at the commencement of the project, they were allowed to adjust that list in light of the experience they had gained with the project. In fact, they were allowed to add user stories if they wished. Once a tentative list of Milestone 2 user stories was selected, as with Milestone 1 these users stories were decomposed into tasks and estimated, with the set of user stories modified as needed to meet a specified total time range for the milestone. Finally, Milestone 2 was presented to the instructor for review and, perhaps after some negotiation, approval. Milestone 3 could be handled similarly, although as the course has been offered it has been less formal, for reasons described below.

3.4 Course Structure

The course met for 50 minutes once per week. The first four weeks of the semester were lecture format covering a different set of topics each week, such as Scrum and Git/GitHub concepts, automated testing, and xdoc commenting.

The remaining weeks were devoted to student presentations. In the early going, three students presented per week, while in the latter half—when the projects had more material to review—there were two presentations per week. Students were required to evaluate the presentations by their peers. To encourage thoughtful evaluations, students were graded, in part, on the instructor's subjective sense of the quality of their evaluations (more on grading later).

For the first set of reviews, the students—and instructor—evaluated several aspects of the presentation, such as how clearly and completely the project, the Milestone 1 user stories, selected Milestone 1 tasks, and the high-level architecture were described. Going beyond presentation, students were asked to review the quality of the underlying work: Do task titles make clear what work must be done, is the system architecture suitable for the project, does the code use good variable names and include helpful comments, does the code appear to be of high quality? Finally, students were asked to compare the project under review with their own work to date, asking, for instance, what they might be able to improve in their own work based on their peer's example.

The second set of reviews were based on the assumption that students were at least well on their way to achieving the first milestone. Thus, review topics now included user acceptance tests, automated test code and coverage, and demonstrated functionality of the system. All of this student-generated feedback was made available to the presenters after their presentations. However, such feedback only affected a student's grade to the extent that it influenced the instructor's own view of the work.

3.5 Grading

All students were required to log and have approved by the instructor 100 hours outside of class to receive a passing grade for the course. In addition, a small portion of the grade was based on consistency of effort, which encouraged reasonably consistent amounts of work being done by most students throughout the semester and facilitated continuous integration.

In keeping with several of the instructor's objectives for the course—namely, reasonably objective assessments and encouragement for good-quality artifacts and reviews—only 25% of a student's course grade was based on their producing functional project code. Other graded elements included attendance (since peer reviews occurred during class) and the quality of not only code but also testing, documentation, presentations, and peer reviews. Of the functionality portion of the course grade, 15%, all or nothing, was earned by achieving the first milestone, with 10% allocated for achieving the second. Students were told from the outset that user stories for the third milestone would not count directly toward their grade but instead would guide the work of students who achieved the first two milestones before the end of the course and who still needed to log 100 hours. This grading structure meant that, with 90% being an A- grade, a student presumably needed to complete both milestones to earn better than an A- for the course (although in practice, this instructor allowed partial credit for the second milestone). This grade structure also meant that a student who did not complete even the first milestone could still potentially pass the course as long as they scored well enough on all other graded elements.

4 Student Feedback

Course feedback gathered by the author's institution was quite favorable. Six of nine students completed the course survey for the first course, seven of 11 for the second. The overall course rating was 4.94/5 for the first course. Although most of the questions on this survey were worded as rating the instructor, this average was noticeably above the norm for this instructor. The survey for the second course asked more specifically about the course itself. Students indicated that they understood what was expected of them (mean score 4.43/5; one was neutral on this question), had opportunities to think critically or in new ways (4.71/5, all agreeing or strongly agreeing), found the course reasonably (4) or too (3) challenging (all of the too-challenging students had cumulative GPAs below 3.5), and rated the learning experience 3.14/4. The lone written comment regarding the learning experience perhaps provides some context: "[The instructor's] strict review of my code made this course very challenging and frustrating. I am glad I was challenged, but I [sic] overall this was a very stressful course." All of the second-course scores were above—in the case of critical thinking well above—broader averages.

Other, context-free, written comments were also encouraging. For instance, consider this comment that suggests an appreciation for being enculturated: "I thought using Github [sic] was really helpful because I've never used it before so it really helped me become more confident using it. Additionally, the project was very helpful to get used to the flow of work, get used to writing documentation and test code, as well as becoming comfortable with having other people reviewing, looking at, and evaluating my code." And the following comment seemed to validate the approach to in-class time and reviews: "I like the presentations and I like the accountability."

References

[1] M.A.C. Clark and R.D. Boyle. "A Personal Theory of Teaching Computing Through Final Year Projects". In: *Computer Science Education* 9.3 (1999), pp. 200–214. DOI: 10.1076/csed.9.3.200.3801.

[2] Hossein Hassani et al. "Supervision of Undergraduate Final Year Projects in Computing: A Case Study". In: *Education Sciences* 8.4 (2018). DOI: 10.3390/educsci8040210.

[3] Heath A. James, Kenneth A. Hawick, and Christie James. "Teaching students how to be Computer Scientists through student projects". In: *IFAC Symposium on Advances in Control Education*. 2005.

[4] B. Olsson et al. "Running Research-Oriented Final Year Projects for CS and IS Students". In: *SIGCSE Bull.* 35.1 (Jan. 2003), pp. 79–83. DOI: 10.1145/792548.611938.

[5] Ken Schwaber. "SCRUM Development Process". In: *Business Object Design and Implementation*. Ed. by Jeff Sutherland et al. London: Springer London, 1997, pp. 117–134. ISBN: 978-1-4471-0947-1.

[6] Hirotaka Takeuchi and Ikujiro Nonaka. "The new new product development game". In: *Harvard Business Review* 64.1 (1986), pp. 137–146.

Thinking Inside the Box: Using Containers To Encourage Commit Discipline*

Robert Marmorstein
Department of Mathematics and Computer Science
Longwood University
Farmville, VA 23909
marmorsteinrm@longwood.edu

Abstract

Integrating “DevOps” (Developer Operations) practices into existing courses can be a convenient way to provide students with valuable career preparation without taking time away from other important course material. In particular, the use of containers to automate application deployment is widely used in industry to deployment of applications to the cloud, but can also allow students and faculty to spend more time on content and less time installing and configuring project software. In our operating systems course, which uses containers to streamline delivery of the PintOS instructional operating system, we discovered an additional advantage: integration of the container environment with the source control system encourages students to commit more frequently and use better software hygiene.

1 Introduction

Asking students to complete large projects, especially large group projects, is a common feature of software engineering, operating systems, web programming, and other upper-level courses. Often, one goal of these projects is to encourage students to develop good habits, such as writing unit tests for each module, properly documenting code, or effectively using a version control system (such as git or mercurial). These skills prepare students for other classes and for effective collaboration in the workforce. The practices collectively know as "DevOps" (developer operations) are particularly important for career preparation. This includes training in source code management, configuration management, continuous integration, testing/test-driven development, and containerization.

In a survey of thirty-nine institutions, Pang, Hindle, and Barbosa [11] found that, other than testing, computer science programs provide only limited exposure to these skills. Demchenko, et al. [6], discuss the proper place for these topics in a computer science program and present a course specifically dedicated to developing these skills. However, they find that most programs that cover these skills, introduce them as secondary topics in existing courses.

Operating systems courses that use projects based on an instructional operating system (such as NachOS [3], PintOS [12], PANDOS [8], OS/161 [9], or Minix [16]), can be a natural fit for introducing some of these skills. These projects tend to be larger and more complex than assignments from other classes, so the benefits of DevOps techniques (such as source code management, containerization, continuous integration, and test-driven design) are easier to motivate.

One particularly desirable outcome is for students to develop good "commit discipline". Learning to commit code often and leave brief, descriptive commit messages in the change log leads to better collaboration, a cleaner development history, and fewer merge conflicts and other problems.

2 Using Docker with PintOS

At Longwood, our operating system course uses the PintOS [12] project. An advantage of PintOS is that it includes an excellent set of test cases for each project. Students (or faculty) can clone and modify these relatively easily (this encourages test-driven design). The most prominent difficulty in running PintOS is that it requires a very specific development environment. In particular, while it can run in the Bochs emulator, some features require a patched version of the QEMU [1] emulator to function correctly.

Keeping this environment installed and working in our laboratory environment required a considerable amount of effort each time we ran the course and made it difficult for students to use at home or on personal machines. To solve this, we created a Docker image that automates these tasks. A version of the scripts for building this image are available at `https://gitlab.com/atomopawn/pintos-docker.git`. We have made some changes so that it can be used outside our lab environment.

Docker [10] images are relatively simple to construct. The user creates a build script called a Dockerfile and uses the docker build command to convert it into an image. Once the image has been created, it can be launched with the docker run command to create a container environment with an independent file system, process space, and network configuration. Applications run inside the container are isolated from the rest of the operating system, which provides security advantages and avoids dependency conflicts with other software.

This approach is also used at the University of Toronto at Scarborough [13] and an alternative Docker image for compiling PintOS is available through the Docker Hub [15] (though it hasn't been updated for some time).

```
FROM debian:bookworm as devbase
RUN apt-get -y update
RUN apt-get -y upgrade
RUN apt-get -y install <packages>

RUN useradd -m pintos

...

USER pintos:pintos
WORKDIR /home/pintos

RUN git config --global pull.rebase false

#Change this to your own email address
RUN git config --global user.email "pintos@example.com"

#Change this to your own name
RUN git config --global user.name "PintOS␣Developer"

#Change this to point to your own PintOS fork
RUN git clone git://pintos-os.org/pintos-anon pintos

COPY bashrc /home/pintos/.bashrc
COPY bash\_logout /home/pintos/.bash\_logout
COPY vimrc /home/pintos/.vimrc

WORKDIR /home/pintos/pintos/src/utils

...

USER pintos:pintos
WORKDIR /home/pintos/pintos/src
CMD /bin/bash -l
```

Listing 1: PintOS Dockerfile

Part of the Dockerfile we use for PintOS is shown in Listing 1. It provides a Debian Linux base system, a compiler and other development and debugging tools, a properly patched version of the QEMU emulator, and scripts for compiling, running, and testing the PintOS projects.

Missing from the figure are the specific list of Debian packages installed to set up the development environment, configuration steps for setting up authentication to a git server over SSH, and commands for compiling the Bochs virtual machine from source.

The Dockerfile also installs two important startup scripts: a bashrc script (Listing 2) that is executed every time a student enters the docker environment and a bash_logout script (Listing 3) that is executed when the student logs out.

3 Student Workflow

When using Docker, students have to navigate two environments: the environment provided by the container and the operating system of the host system. One challenge this creates is that teams need a way to access their source code within the container environment. The most elegant way to do this is probably the use of "volumes" to make a folder from the host environment inside the container, but user permissions and the design of our lab environment make this unworkable.

```
git config pull.rebase false
git checkout -b work
git pull --set-upstream origin work
git push --set-upstream origin work
```

Listing 2: The .bashrc file

```
cd /home/pintos/pintos
git commit -a; git push
```

Listing 3: The .bash_logout file

Instead, we give each team of students a git repository on a private git server. The git repository includes a copy of the pintos code, but also contains a "build" folder containing the Dockerfile and the two bash scripts. We provide a README file with instructions for modifying the Dockerfile and rebuilding the image.

The .bashrc script automatically pulls code from the work branch of this repository into the container every time the students run it. The .bash_logout file ensures that students are prompted to commit and push any changes they make while working inside the container before it closes. They can accept these changes by saving and quitting the editor, or quit without saving to discard them.

To make this work, credentials for committing to the git repository have to be added to the Docker image. Students clone the git repository into a local folder, edit the Dockerfile to set their name, e-mail address, and the URL of the git repository and then add public and private key files for SSH authentication to the server.

Following these configuration steps, students can build the docker container by typing `docker build -t pintos .` and they can launch the container environment using `docker run --rm -it --name pintos pintos`.

This means that the typical workflow is for students to work outside of the container, commit their code to git, then build and run the container to test their work. If a test fails or the team wants to run the code in a debugger, they can work directly from the container.

This design at first seemed clunky and awkward and at least one group failed to follow instructions properly, which resulted in some confusion when changes by one student showed up in the repository under another student's name and e-mail address. Students adapted to this workflow surprisingly quickly, however, and by the end of the first project most of them were using it confidently and effectively. Furthermore, we noticed a distinct and surprising advantage of this solution: it taught students to commit changes often and to use good commit discipline.

4 Commit Discipline

Students are often told to "commit early and commit often", but good commit discipline can be challenging to encourage when class projects are focused on broader learning objectives. In introductory courses, where project complexity is small, it can be difficult for students to see the advantage of frequent commits. For example, Conner, McCarthy, and Lambert [5] describe a CS1 course in which they introduce students to git, but their conclusion is that students manage the cognitive load better if these skills are taught later in the major. In more advanced courses, it is difficult to balance time spent on core topics of the course and time spent teaching software tools.

In the case of group projects, it is possible to introduce source control as a means of collaboration. However, it is not always obvious to students why they should commit frequently until merging code results in merge conflicts. In our experience, when this occurs, students often blame the source control software instead of changing their commit practices.

A common approach to this problem is to introduce version control in a sophomore-level data structures and algorithms course, where projects have reasonable complexity, but students have already developed facility with the programming language. This approach is used by Eloe [7], who argues that students are not ready for these skills in an introductory class, but that introducing them at an intermediate level allows students to use and refine these skills in further courses. To address this problem, Eloe proposes a milestone-based approach in which class projects are divided into smaller checkpoints and students are required to commit specific changes at regular intervals. The use of milestone commits also enables him to introduce students to test driven development and the idea of continuous integration.

Even at this level, however, teaching good software hygiene takes course time away from other topics. In order to introduce this subject as efficiently as possible, Berg, Osnes, and Glassey [2] applied a scaffolded project approach in their data structures and algorithms courses. In their initial project, they require students to make only three git commits, but then add additional quality requirements for subsequent submissions. They support these changes by lecturing on the importance of commit quality and how to maintain good commit discipline. They found that, on average, commit frequency increased (over a control group) with each project.

In both of these approaches, the motivation for students to commit frequently comes from the project structure and is imposed as an additional project requirement. An alternative idea is to incentivize frequent commits by tying source control to the project grade in a less direct way. For example, the gitkeeper [4] tool integrates source control with project submission. Instead of turning in projects as a lump sum, students use git to submit multiple revisions. The tool runs automated tests on each submission and gives students feedback. Since students can improve their grade by submitting multiple times, they have a strong incentive to commit early and often. This approach works well if a set of automated tests is available, integrated into the tool, and the project structure allows it to be incorporated into the project grade.

Requiring students to commit in order to build their projects had a similar effect. Instead of committing code only when they want to share it with others, students commit every time they want to compile and test. For the first project, the average commit frequency was 36.7 commits per week. This increased to 40.3 commits per week by the end of the second project - a modest improvement.

5 Conclusion

Our experience using Docker in the operating systems course has been largely positive and we received lots of positive feedback from the students about it.

While commit frequency was noticeably excellent for these projects, students at first had a tendency to use short, non-descriptive commit messages. Some examples from the first project are shown in Table 1.

While it is difficult to measure improvement in this area, anecdotal evidence

Commit Message
help
commented out something
whats poppin
i did a dumb
working

Table 1: Student commit messages from the first project

suggests that this had improved by the end of the semester. Some examples from the last project are show in Table 2. Clearly, this is an area in which additional instruction might still provide improvement.

Commit Message
added the proc_info struct to the thread.h
moved copy_string_read above setting return value to eax in read
ridded a pest
added stuff that doesnt work
Allocate page for file_map, add page variable to file_map to store page to be freed.

Table 2: Student commit messages from the last project

One drawback to using containers is that connecting a debugger to the virtual machine requires one additional step. PintOS provides a debugging interface through gdb's "remote debugging" system [14]. This requires the use of two terminals: one to run the code and another to run the remote debugger. When a student runs pintos in debug mode inside the container, gdb cannot access it directly from the host system. A solution to this problem is to use the "docker exec" command to open a second shell within the container. Students can type `docker exec -it pintos /bin/bash` to create a second terminal for debugging.

Anecdotally, one notable difference from previous offerings of the course was that students experienced far fewer merge conflicts. With only one exception, they were able to resolve them on their own without intervention from the instructor. One group did experience significant problems with merges. They had modified the Dockerfile to pull from different branches of the git repository and wound up with two incompatible solutions to a project. Attempts to merge these two solutions created more problems and the instructor had to intervene to get the group back on track. Perhaps not surprisingly, subsequent development by this group showed remarkably improved commit discipline.

References

[1] Fabrice Bellard. “QEMU, a Fast and Portable Dynamic Translator”. In: *Proceedings of the Annual Conference on USENIX Annual Technical Conference*. ATEC ’05. Anaheim, CA: USENIX Association, 2005, p. 41.

[2] Amanda Berg, Simon Osnes, and Richard Glassey. “If in Doubt, Try Three: Developing Better Version Control Commit Behaviour with First Year Students”. In: *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education - Volume 1*. SIGCSE 2022. Providence, RI, USA: Association for Computing Machinery, 2022, pp. 362–368. ISBN: 9781450390705. DOI: `10.1145/3478431.3499371`. URL: `https://doi.org/10.1145/3478431.3499371`.

[3] Wayne A. Christopher, Steven J. Procter, and Thomas E. Anderson. “The Nachos Instructional Operating System”. In: *Proceedings of the USENIX Winter 1993 Conference Proceedings on USENIX Winter 1993 Conference Proceedings*. USENIX’93. San Diego, California: USENIX Association, 1993, p. 4.

[4] Ben Coleman and Nathan Sommer. “Git-Keeper: Streamlined Software for Automated Assessment Workflows”. In: *J. Comput. Sci. Coll.* 38.3 (Nov. 2022), p. 177. ISSN: 1937-4771.

[5] David C. Conner, Matthew McCarthy, and Lynn Lambert. “Integrating Git into CS1/2”. In: *J. Comput. Sci. Coll.* 35.3 (Oct. 2019), pp. 112–121. ISSN: 1937-4771.

[6] Yuri Demchenko et al. “Teaching DevOps and Cloud Based Software Engineering in University Curricula”. In: *2019 15th International Conference on eScience (eScience)*. 2019, pp. 548–552. DOI: `10.1109/eScience.2019.00075`.

[7] Nathan W. Eloe. “Teach like a Git: Streamlining Assignment Administration and Enforcing Good Habits with Professional Tools and Software Development Practices”. In: *J. Comput. Sci. Coll.* 36.6 (Apr. 2021), pp. 27–36. ISSN: 1937-4771.

[8] Michael Goldweber and Renzo Davoli. “Student Guide to the PANDOS Project”. In: (2020). URL: `https://wiki.virtualsquare.org/education/doc/pandos.pdf`.

[9] David A. Holland, Ada T. Lim, and Margo I. Seltzer. “A New Instructional Operating System”. In: *SIGCSE Bull.* 34.1 (Feb. 2002), pp. 111–115. ISSN: 0097-8418. DOI: `10.1145/563517.563383`. URL: `https://doi.org/10.1145/563517.563383`.

[10] Solomon Hykes et al. “Docker”. In: *URL: https://www. docker. com* (2015).

[11] Candy Pang, Abram Hindle, and Denilson Barbosa. "Understanding Devops Education with Grounded Theory". In: *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Software Engineering Education and Training*. ICSE-SEET '20. Seoul, South Korea: Association for Computing Machinery, 2020, pp. 107–118. ISBN: 9781450371247. DOI: 10.1145/3377814.3381711. URL: https://doi.org/10.1145/3377814.3381711.

[12] Ben Pfaff, Anthony Romano, and Godmar Back. "The Pintos Instructional Operating System Kernel". In: *SIGCSE Bull.* 41.1 (Mar. 2009), pp. 453–457. ISSN: 0097-8418. DOI: 10.1145/1539024.1509023. URL: https://doi.org/10.1145/1539024.1509023.

[13] Thierry Sans. *CSCC59 Pintos Instructions*. URL: https://thierrysans.me/CSCC69/projects/WWW/pintos.html.

[14] Richard M. Stallman, Roland Pesch, Stan Shebs, et al. *Debugging with GDB*. URL: https://sourceware.org/gdb/current/onlinedocs/gdb#Remote-Debugging.

[15] John Starich. *Pintos for Docker*. URL: https://hub.docker.com/r/johnstarich/pintos/.

[16] Andrew S Tanenbaum. "Lessons learned from 30 years of MINIX". In: *Communications of the ACM* 59.3 (2016), pp. 70–78.

Enhancing Level and World Design in Game Development through Data Analytics[*]

Gabriel Loring, Michelle Liu
School of Technology and Innovation
Marymount University, Arlington, VA 22201
{gxl93136, xliu}@marymount.edu

Abstract

While data analytics plays a crucial role in improving game design, there are gaps in the existing literature regarding their specific applications to areas such as level design and world design. This study aimed to address such gap by examining the effect of qualitative and quantitative data analysis methods, specifically content analysis and linear regression, on level design and world design. The data for analysis were collected from demonstrations programmed in Unity with the intent of having participants record their gameplay. Surveys were utilized as an additional data collection tool. The findings demonstrated that qualitative analysis contributed to improvements in both level and world design. Furthermore, linear regression enabled developers to identify trends for enhancing level design, although it proved less effective for world design. The study acknowledged limitations and proposes future research directions to further explore this subject.

1 Introduction

The purpose of this study is to gain a better understanding of how data analytics can effectively contribute to the processes of level and world design in the field of game design. While data analytics has been discussed in the realm of game design, there has been little focus on its practical application in improving the aforementioned design aspects. Therefore, our research aims to take a closer look at various analytical methods to enhance the level and world design of a game as a means of an exploratory study in this area.

The gaming industry is widely recognized as one of the most innovative and dynamic sectors in the field of technology, influencing culture, social networking and entertainment. Many attentions have been paid to design practice. However, using data analytics to improve specific aspects of game design has received relatively little discussion. This gap primarily arises from the predominant focus of data analytics on business-oriented objectives, such as player retention and financial profitability. Consequently, this study aims to address this gap by delving into an internal perspective of gaming design, with the objective of uncovering insights that can improve the intrinsic quality of games rather than solely on their financial success post- release.

To facilitate the playtesting process, two functional game demonstrations were developed, each targeting a specific aspect of game design. The first demonstration focused on level-based design and was programmed as a two-dimensional platforming game inspired by the renowned Mario series. The second demonstration focused on world-based design and was programmed as a two- dimensional, top-down perspective reminiscent of games in the Final Fantasy series. Both demonstrations were provided to a select group of playtesters, who were subsequently asked to complete a survey comprising Likert Scale items and open-ended questions. This study employed both quantitative and qualitative measures through data analysis to explore avenues for improving level-based and world-based design.

2 Background

Previous studies have investigated the application of data analytics in game design, but often with a broader scope [5]. In contrast, this research focuses on the narrower aspects of game design, namely level design and world design. However, it is still valuable to acknowledge and draw upon the existing research, as it provides a foundation for conducting data analysis in this study.

One prior study focused on analyzing player behavior in World of Warcraft to determine the probabilities of subscription renewal by using time-related metrics such as player activity and playtime [4]. We found that tracking the duration of playtests used by [4]. was critical for conducting effective analysis in our research. Regression analysis, stated to be the one of the most common models, was also used in this study through linear, multiple linear, and curvilinear regressions to discover trends in the data using response and predictor variables. Understanding these regression techniques was essential for our chosen method of data analysis. One notable framework called Games and Learning Analytics for Educational Research (GLEANER) is a process devised for supplementing tracking and analyzing behaviors of players during gameplay [1] However, it is important to note that the implementation of GLEANER requires careful consideration during the game design phase.

Moreover, other studies explored valuable aspects relevant to our research which provides insights into metrics and considerations for game design [2, 3]. For example, one study focused on the significance of difficulty in educational games, emphasizing the need to balance learning, enjoyment, and progression [3]. Understanding the dynamics of difficulty is crucial for designing engaging game environments. Additionally, the articles highlighted the importance of identifying potential stopping points due to excessive difficulty and incorporating measures of player engagement through surveys, which were also incorporated into our research as valuable metrics for data collection.

3 Research Methodology

3.1 Rationale for the Research Approach

To explore the utility of data analytics in level design and world design, the primary approach employed was playtesting. Participants were provided with a copy of the game demonstrations and instructed to engage in gameplay to the best of their ability. Subsequently, the participants were given a survey to complete. The survey was designed to elicit more nuanced critique of the design and generate insights that can be deemed quantifiable like difficulty or level of experience with the game style.

In the level design demonstration, players were to navigate platforms, jump over enemies, and avoid traps in order to reach a token at the end of two levels. The world design demonstration focused on tasking the participants with the goal of reaching an inn in a small town. To get into the inn, players had to complete all three provided tasks that directed them to the other parts of the game area. In the second iteration, additional, but optional, tasks called sidequests were introduced, providing participants with extra goals beyond the major objectives.

3.2 Sampling and Sampling Procedure

A convenience sampling strategy was utilized for the survey, targeting individuals with video game playing experience. Participants for the survey were recruited from Discord servers, where a brief project overview was posted. Interested individuals were instructed to send a direct message for further information. The inclusion criteria required participants to be willing to download the game demonstrations, provide an email address, and allocate time for data collection. Due to time and recruitment constraints, the sample size for the first iteration was ten participants.

3.3 Procedure for Recruitment, Participation, and Data Collection

The Institution Review Board (IRB) at the authors' University reviewed and approved this research according to the 45CFR46.101(b)(2): (2) Tests, Surveys, Interviews.

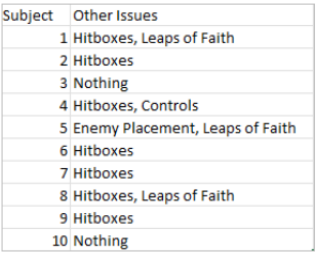
Participants were recruited through Discord, with project descriptions posted and interested individuals prompted to send direct messages. Informed consent was obtained via an online form. Participants downloaded the game demonstrations and recorded their gameplay experiences, which were reviewed for accuracy and entered into a spreadsheet. They were then given an online survey to provide feedback, which was also logged into the spreadsheet. After analyzing the data and making necessary adjustments, a second iteration was conducted with updated demonstrations and a revised survey. More than half of the participants agreed to participate in the second iteration, and the data collected from it was included in the analysis.

4 Results

The time frame for data collection during the first iteration of playtesting was approximately one week, beginning with the recruitment process and ending with the survey completed. The second iteration in turn took only a couple days as the initial recruitment process was not necessary to conduct. Data primarily focused on playtest metrics, with additional responses addressing descriptive aspects such as difficulty perception and participants' experience with similar games. The sample size of ten participants was small but sufficient for the scope and nature of this project, which aimed to explore the utility of data analytics on user feedback in level and world design. To address this question effectively, it was necessary to break it down into smaller sub-questions for organizational purposes. Median was utilized as a measure of central tendency considering the small sample size.

4.1 What is the Effect of Qualitative Data?

The qualitative responses obtained from the surveys provided valuable insights for improvement in this project. Although the majority of the data consisted of simple yes/no or preference-based answers, as well as observations of participant behaviors, it was important to consider that this qualitative data, along with the "Difficulty for Enjoyment" section, guided the improvement process for both level and world design. Despite the overall uniformity of the data, it still played a crucial role in informing necessary refinements. Notably, there was a section of qualitative data that deviated from the uniform pattern observed in the rest of the responses, as shown in Figure 1.



Subject	Other Issues
1	Hitboxes, Leaps of Faith
2	Hitboxes
3	Nothing
4	Hitboxes, Controls
5	Enemy Placement, Leaps of Faith
6	Hitboxes
7	Hitboxes
8	Hitboxes, Leaps of Faith
9	Hitboxes
10	Nothing

Figure 1: Exception to Uniform Qualitative Data

Figure 1 presents participants' responses to a question regarding resets and other issues encountered during the level design playtest. Through content analysis, the responses were categorized into specific themes, highlighting the common concerns among participants. While two participants reported no issues, seven participants expressed difficulties with hitboxes of the player character and enemies, making it the most prevalent issue. The next notable concern was related to "leaps of faith," referring to jumps where the landing point is not visible, requiring blind jumps. Valuable insights were gained by extracting and condensing the survey responses into themes, which facilitated the developers to identify areas that require improvement.

Despite its overall uniformity, the qualitative data collected in this study remains crucial for improving the level and world design demonstrations. Specifically, in relation to the "Level 1 Wrong Turn" item, it pertains to a section of the level involving a challenging set of jumps where players often took the wrong path, causing confusion and delaying progress. To address this issue, the second iteration incorporated arrows to provide clearer direction, as depicted in Figure 2. Thus, by considering and analyzing the uniform qualitative data, valuable adjustments were made to enhance the gameplay experience.

In addition, a split in participant responses prompted a deeper consideration and evaluation of the difficulty in the level design. This presented a challenge for the developers in determining how to make changes to the demonstration, deviating from the original intention of following the majority's preferences. Ultimately, the decision was made to adjust the levels to be easier, which had a direct impact on subsequent modifications.

All participants agreed that the environment plays a significant role in enhancing their engagement with a game in the world design demonstration.

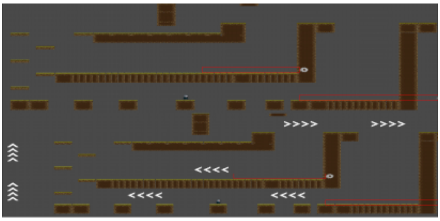


Figure 2: Before and After of Section Due to Wrong Turns

Although open to interpretation, this feedback led to a focus on addressing the relatively barren areas in the original environment. Simple additions such as trees helped to fill these spaces and contributed to increased engagement between iterations for participants who took part in both phases. Figure 3 shows a comparison of grassland area in the first and second iteration. Several environments in the world design demonstration remained unchanged in the first iteration of playtesting. However, considering the responses received, it is noteworthy that four out of six participants reported an increased level of engagement in the second iteration compared to the first. It is important to acknowledge that while the addition of scenery is not the sole factor contributing to this perceived improvement, it can be interpreted as having an influence on the participants' overall experience.

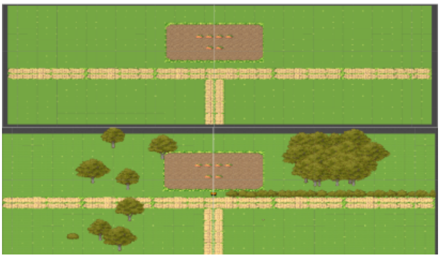


Figure 3: Comparison of First and Second Iteration of Grassland Area

Taking into account the responses shown in Figure 1, it becomes evident that the most prevalent issue reported by participants was related to hitboxes. Qualitative analysis indicated that addressing this issue was of utmost importance. Video recordings further supported participants' observations, revealing instances where players would unintentionally land on enemies, resulting in their death and subsequent level reset, contrary to the intended interaction of defeating the enemy. As shown in Figure 4, adjustments were made to the colliders of both the player and enemy. The player's lower collider was enlarged to ensure successful enemy destruction upon landing, while the enemy hitbox was increased in size. These modifications, in conjunction with other factors, contributed to a reduction in deaths during the second iteration. However, it should be noted that these adjustments alone do not fully account for the observed decrease.

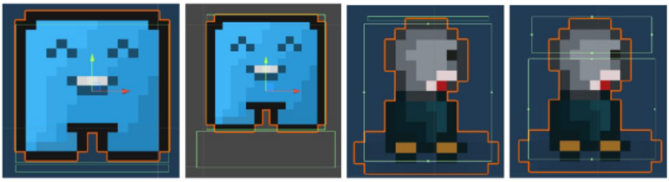


Figure 4: Player Character (left) and Enemy (right) Hitboxes, before and after

In summary, the qualitative data collected, despite its uniform nature, proves to be valuable for driving improvements in both level and world design. It not only enables general enhancements but also provides insights into specific areas, such as participants' preferred difficulty, which opens up avenues for targeted improvements in level design. However, in cases where the qualitative data deviated from uniformity, it offered valuable insights into issues that may not have been adequately captured by quantitative data. Content analysis highlighted the most prominent issue, which, in this instance, was the hitboxes.

4.2 What is the Effect of Linear Regression in Level Design?

Linear regression analysis was employed to examine the relationship between variables in the collected data from the first iteration of playtesting in the level design demonstration. Upon examining the p-values, it was observed that each regression yielded a value below 0.01 or 0.05, indicating a statistically significant and robust linear relationship between the variables. Due to space limitations, a subset of graphs and findings was chosen to be included, providing a focused representation of the results.

The scaling of the graphs in Figure 5 may have seemed off due to clustering of values, making it difficult to track trends. However, the equation in the lower right corner served as a good indicator. It showed that Level 1-2 exhibited the least increase in resets when predicted against a longer time in Level 1. Additionally, the formula for Level 1-1 showed the largest correlation as time went on, which made sense since each reset required players to start the level from the beginning, increasing the likelihood of resetting in the first section.

Figure 6 showed a similar trend to the linear regressions in Figure 5, with

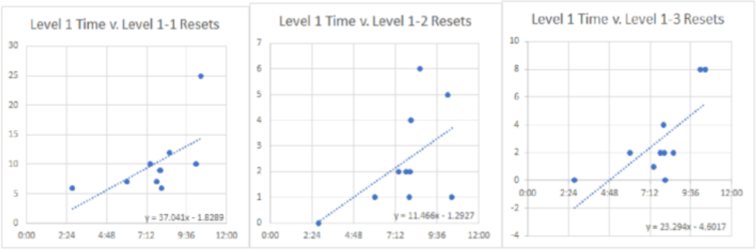


Figure 5: Level 1 Time v. L1-1, L1-2, and L1-3 Resets for Iteration One

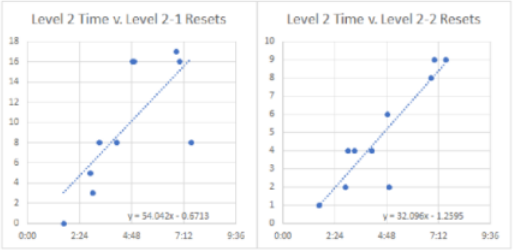


Figure 6: Level 2 Time v. L2-1 and L2-2 Resets for Iteration One

the first section in the level demonstrating a higher correlation. Despite the visual scaling, this fact can be considered when determining improvements based on the desired difficulty. Further analysis can be conducted on Levels 1 and 2, specifically regarding resets as a combination of deaths and falls, to gain additional insights.

Figure 7 provided insights into the correlation between deaths and resets in specific sections. For Level 1-1, the graph for deaths mirrored the graph for resets, as falls were not considered. This allowed us to accurately analyze the correlation associated with deaths. Figure 8 highlighted that in Level 1-1, where falls were absent, the trendline was flat, indicating no variance. However, in Level 2-2, falls played a significant role in the correlation between the time spent in Level 2 and the resets in Level 2-2.

The goal was to make the level easier for participants, considering factors such as enemy placement, enemy speed, platform placement, trap speed, and field of view. Changes were made to the player's field of view and hitboxes based on qualitative data analysis. Specific adjustments were also made to address the contribution of falls in Level 2-2, particularly in the jump at the start of the section. Visual changes, depicted in Figures 9 and 10, included an extended camera view for better visibility and the addition of platforms to

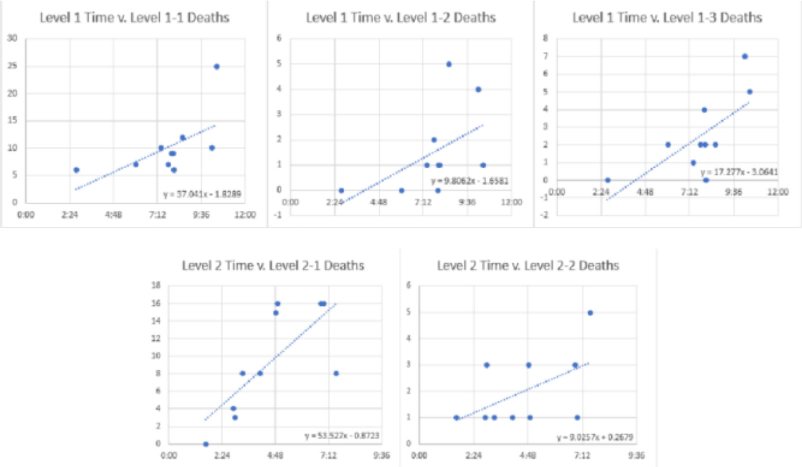


Figure 7: Level 1 Time v. L1-1, L1-2, and L1-3 Deaths for Iteration One (top), Level 2 Time v. L2-1 and L2-2 Deaths for Iteration One (bottom)

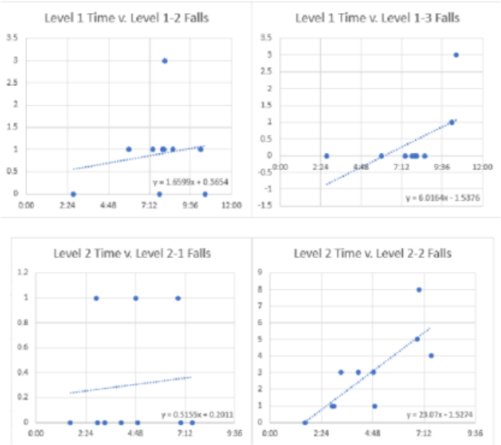


Figure 8: Level 1 Time v. L1-1, L1-2, and L1-3 Falls for Iteration One (top), Level 2 Time v. L2-1 and L2-2 Falls

assist players in progressing through the level.

The linear regression showcased the positive effects of hitbox adjustments and an expanded field of view, leading to a lower correlation between deaths and times in respective levels. Future improvements could focus on striking

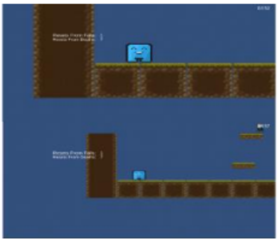


Figure 9: Field of View Comparison, Iteration One (Top), Iteration Two (Bottom)

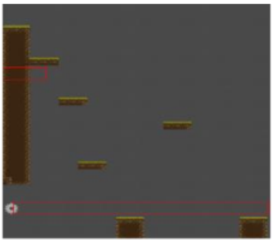


Figure 10: Section 2-2 Jump in Iteration Two

a balance between increasing difficulty and maintaining player engagement. The metrics collected from the level-design game alone were not the only ones to consider; metrics such as the participants' Difficulty Score provide valuable insights when examined independently, but require further analysis when dissected into smaller components.

Data analysis employed in this study proves valuable for comparing different aspects of level design and predicting variable trends as time progresses. Additionally, it helps identify metrics that may not contribute significantly to level design progression. It is important to note that graph scaling can influence the perception of correlation strength, emphasizing the need to consider relationships between data values. By leveraging these insights, informed choices can be made to directly impact the design improvements in subsequent iterations.

4.3 What is the Effect of Linear Regression in World Design?

The process for utilizing linear regression in world design improvement is similar to the previous question, but with less extensive explanation.

Figure 11 revealed a weak correlation between Time and Engagement Level, as well as a weaker correlation between Time and Puzzle Engagement, suggest-

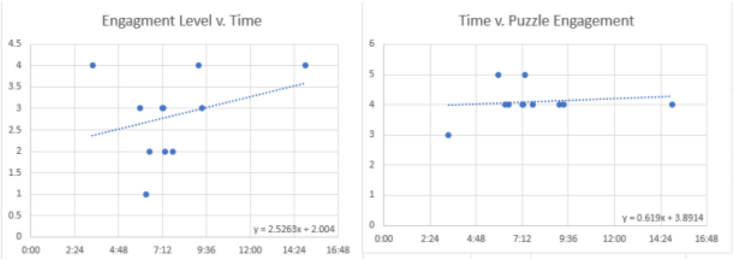


Figure 11: World Design Time v. Engagement Level and Puzzle Engagement

ing limited association between time spent and participants' engagement levels or the impact of puzzles on engagement.

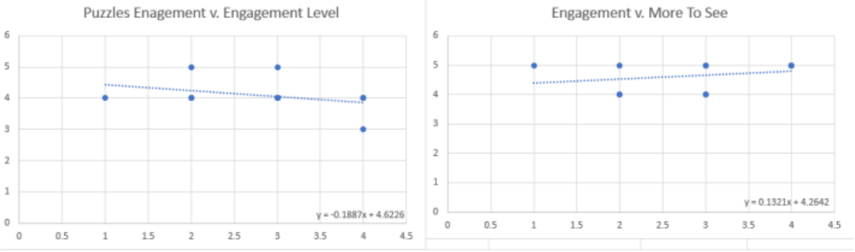


Figure 12: Engagement Level v. Puzzle Engagement and Desire to See More

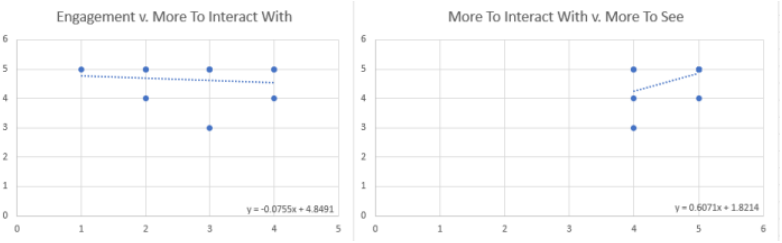


Figure 13: Engagement v. Desire to Have More Interaction and Desire to Have More Interaction v. Desire to See More

Figures 12 and 13 revealed no significant correlation in all four linear regressions, suggesting a lack of strong associations between the variables. This can be attributed to the uniformity of participants' opinions on the playtest and their preferences for a game focused on world design.

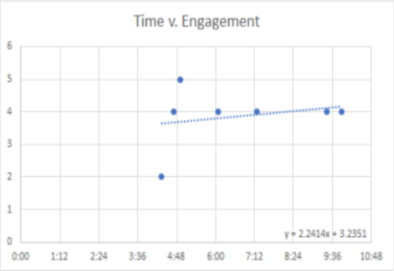


Figure 14: Time v. Engagement Level for Iteration Two

As shown in Figure 14, very little has changed in regards to the correlation between Engagement and Time spent playing the demonstration, even after including additional sidequests. It led to the conclusion that the sidequests had no significant impact on engagement or time. It should be noted as well that the participants during this iteration have known the solutions as no changes were made from the first iteration. The lack of correlation between any of these variables resulted in the fact that linear regression is not sufficient for generating deeper insight for improvement between iterations for world design.

5 Discussion

Qualitative data analysis helped the researchers understand the nuances and subjective aspects of level and world design based on feedback going beyond numerical metrics. It allowed researchers to gain better understanding of the participants' perspectives, preferences, and experiences. In this study, qualitative analysis revealed specific areas for improvement, such as the need for more scenic elements in world design or addressing common issues in level design.

In this study, the researchers used linear regression analysis to examine correlations between different metrics and gameplay elements. This type of quantitative analysis helped identify significant factors that influenced players' perceptions of and interaction with game demonstrations, including the duration of gameplay and level of engagement. However, the researchers must highlight that much of the insight on improvements resulted from breaking down specific factors like the levels and where deaths and falls occurred into smaller subsets to reveal where certain aspects had more bearing on the time and in what place. In addition, it is essential to determine through this method what metrics might not be as effective as a viable utility in its own right. Being able to focus on more effective metrics, or even create new ones in their place serves as an effective way to improve the level design process. The effectiveness of linear regression in world design does not particularly provide any new or effective insights based on the collected data. This point must be stressed as more can be done to improve upon this process for future consideration.

Several constraints and limitations required considerations when interpreting the research findings. First, not all participants from the initial playtest were available for the second playtest due to various reasons, and time constraints for data collection period prevented an effective solution. Second, technological constraints, such as antivirus software and participants' technological capabilities, occasionally posed challenges during the playtest. Third, human error also impacted the data, as one participant recorded the wrong screen, resulting in gaps in death occurrences and reliance on available information. These are limitations that cannot immediately be accounted for, but needed to be addressed immediately in order to properly conduct the playtest. For instance, the antivirus issue for one subject was solved by simply having them re-download the playtest with their antivirus software turned off temporarily.

Another constraint worth mentioning is related to the recruitment process. Despite allocating a significant amount of time for recruitment, the number of participants willing to assist in the project ultimately depended on their interest and willingness. In addition, relatively small sample size and drop-off bias due to some participants not assisting for the second iteration may affect the generalizability of the research findings. Lastly, certain survey questions may not have been as conducive to in-depth analysis as initially anticipated.

6 Conclusion

When considering data analytics in level and world design, it is important to acknowledge the significant differences in how data is reacted to and responded to in these two approaches. However, qualitative and quantitative analysis methods, such as content analysis and linear regression, can evidently have an impact on improving level design methodologies. In the case of world design, linear regression analysis did not yield substantial insights when evaluated across multiple metrics. Nevertheless, the qualitative analysis aspect of this project still contributed to improvements, despite some issues with the data results. As discussed, there were limitations in various aspects, which emphasized the need for further exploration of data analytics in the context of level and world design. Precautions should be taken when interpreting the study results due to the limited degree of generalizability. The findings and results drawn from this study are applicable solely to the data gathered for this particular investigation. This leaves ample room for enhancing methodologies and data collection techniques, as well as the opportunity to explore whether the findings may evolve or change.

References

[1] Jannicke Baalsrud Hauge et al. "Implications of Learning Analytics for Serious Game Design". In: *2014 IEEE 14th International Conference on Advanced Learning Technologies*. 2014, pp. 230–232. DOI: 10.1109/ICALT.2014.73.

[2] Drew Hicks et al. "Using Game Analytics to Evaluate Puzzle Design and Level Progression in a Serious Game". In: *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge*. LAK '16. Edinburgh, United Kingdom: Association for Computing Machinery, 2016, pp. 440–448. ISBN: 9781450341905. DOI: 10.1145/2883851.2883953. URL: https://doi.org/10.1145/2883851.2883953.

[3] Yoon Jeon Kim and Jose A. Ruipérez-Valiente. "Data-Driven Game Design: The Case of Difficulty in Educational Games". In: *Addressing Global Challenges and Quality Education: 15th European Conference on Technology Enhanced Learning, EC-TEL 2020, Heidelberg, Germany, September 14–18, 2020, Proceedings*. Heidelberg, Germany: Springer-Verlag, 2020, pp. 449–454. ISBN: 978-3-030-57716-2. DOI: 10.1007/978-3-030-57717-9_43. URL: https://doi.org/10.1007/978-3-030-57717-9_43.

[4] Elton Sarmanho Siqueira et al. "A Data Analysis of Player in World of Warcraft Using Game Data Mining". In: *2017 16th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames)*. 2017, pp. 1–9. DOI: 10.1109/SBGames.2017.00009.

[5] Günter Wallner. *Data Analytics Applications in Gaming and Entertainment*. CRC Press, 2019.

A Secure Real-Time Multimedia Stream Data Backup by Network-Engine for Resource Constrained Devices[*]

Md Amiruzzaman and Ashikahmed Bhuiyan
Department of Computer Science
West Chester University, West Chester, PA 19383
{MAmiruzzaman,ABhuiyan}@wcupa.edu

Abstract

In this study, we implemented a secure network engine for mobile devices. The study focused on faster data communication, the authenticity of the user, data transfer reliability, and data integrity. In addition, we applied advanced security techniques, such as Certification Authority (CA), Message Authentication Code (MAC), logging, and meta-data concepts, to ensure user permission and data access control. Further, a 2-way handshake and real-time data transfer channel were implemented to enhance the communication and data transfer speed. The results of this study show promising results to move forward with high-speed and secure data transfer between mobile devices and server computers.

1 Introduction

Mobile devices are popular among users of all ages due to their mobility features. In most cases, mobility, lightweight, and quick access are the focus of a mobile device. However, due to their size, mobile devices need more resources to store data and run processes. A high-speed internet connection helps to achieve the quick access feature of a mobile device. Also, with the ever-growing digital content landscape, mobile video viewing is expected to become increasingly widespread worldwide.

Recent statistics show that around 76% of the digital video content audience in the United States watches videos on their smartphones [7]. In comparison, 57% of people watch videos on their mobile globally. Meanwhile, 58%, 41%, and 34% of respondents watch videos on a smart TV, laptop, or tablet [7, 4]. Other statistics report that in 2022 there were approximately 4.18 billion mobile internet users worldwide, and a significant portion use mobile to watch videos, leading to doubling mobile video consumption every year.

These statistics indicate that video-based data is prevalent among mobile users, who record and share thousands of video-based data every day. Much video content is available to mobile audiences in the United States, including but not limited to social media videos, fitness platforms, and streaming subscription services. Recent statistics show that around 92% of the videos watched via mobile devices are shared with others [4]. These examples promote the idea of a network-based data backup technique. A network-based data backup technique would allow users to transfer their captured data using the integrated mobile camera and watch videos stored on the remote server. In the last decade (and likely in the next decade), we observed that the capacities of network-based backup techniques continue to grow at over 50% per year [1].

The network-based backup scheme is strongly connected with data protection concerns [6]. As digital technology users, we use many electronic files daily. Often, we access those files for writing, reading, and modifying purposes. Information technology continues to play a vital role in our daily activities. Therefore, we must maintain reliable data storage, backup, and recovery techniques that can help us continue accessing our electronic or digital files. In workplaces, we may hear someone shout, “All my data is gone!! My computer crashed!! I had years of data saved on this computer.” Therefore, it is essential to have a scheme that safeguards electronic data and ensures integrity and availability [12]. However, the transfer time of video-based data through network communication demotivates many users. Therefore, a network-engine-based, fast, and reliable multimedia stream data transfer system is needed.

2 Existing work

Stream-based servers are often used to store and transfer stream data, mainly video-based data. A stream-based back server usually receives data in blocks; therefore, block transfer time is a significant factor that needs to be considered while designing a stream-based backup server [10]. A network protocol divides data into smaller blocks and schedules a smooth transition from client to server. Finally, it reassembles the blocks to construct the data stream necessary to accomplish this task [5]. Most data backup servers use the Transmission Control Protocol (TCP) due to its popularity and packet delivery assurance. However, stream data needs faster transmission and higher throughput, which differs from TCP's focus. Another commonly used protocol to send messages (transported as datagrams in packets) to other hosts on an Internet Protocol (IP) network is the User Datagram Protocol (UDP). The UDP does not require prior communication to set up communication channels. The main drawback of UDP is its unreliability; it does not use any error control mechanism. Also, UDP has no windowing and cannot ensure that data is received in the same order as it was transmitted.

2.1 Challenges in existing work and plausible solutions

The TCP protocol is known for its reliability and ability to detect the missed or droped packets. Furthermore, the TCP protocol uses a 3-way handshake to establish communication in a client-server model, which helps to ensure the reliability of the data transfer. However, many studies [2] have reported the drawbacks of the 3-way handshake, e.g., communication overhead. Therefore, TCP is popular as it ensures data delivery. However, such assurance comes at the cost of communication speed. Hence, the TCP is unfavorable for video or streaming-based data communication, where faster communication is crucial. For this type of communication, the UDP protocol is often used. However, UDP needs more reliability and security aspects of communication.

Finding a better balance between communication speed and reliability is challenging. Sharing data and being able to delegate authority to others is an information security-based problem. While delegation allows an advanced level of sharing data, this process enables a user to authorize different types of permission, such as read, write, and execute. In addition, this process allows a user to add or remove another admin user. Message Authentication Code (MAC) is often used in network-based data transmission, and it ensures the granular level of data sharing and the integrity of the data. In addition, MAC helps to determine any alteration in a transmitted computer file or network packet. However, sending MAC using the regular channel during data communication will add overhead. In addition, the extra information will take up network bandwidth space and time to manage.

2.1.1 Outline of the proposed study

Motivated by challenges stated in section 2.1, this study presents a real-time multimedia stream data backup network engine for resource constrained devices, considering the security aspects. A `network engine` helps to establish a long-term relationship between the client and server-based systems. The network engine will use the Real-time Transport Protocol (RTP) protocol, designed for high-speed multimedia data transmission, such as video data. Further performance improvement was made by implementing a 2-way handshake instead of the 3-way handshake (see section 3.1). The RTP protocol is a partic-

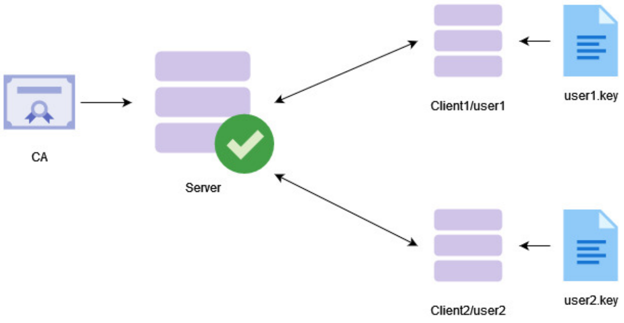


Figure 1: CA-based communication to allow trusted communication

ular type of network protocol that focuses on delivering audio and video over computer networks.

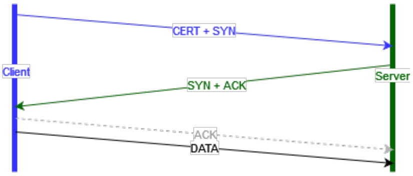


Figure 2: Proposed 2-way handshake. The gray dotted line shows that *ack* and data will start to transmit simultaneously after the client receives the confirmation from the server.

A log-based reliability control and MAC were used to ensure the data's integrity. A Certification Authority (CA)-based model and meta-data techniques were used to satisfy authenticity and rightful access to the data. The CA is a mechanism that allows issuing certificates/fingerprints to clients and servers, which is used to prove identity and ensure authenticity (see Fig. 1). CA also stores, re-issues, and verifies clients and servers.

3 Method

This section presents the steps of this study that helped to develop a faster and reliable network-engine for multimedia stream data transfer, considering access control, delegation, and integrity of the data.

3.1 Implementation details

The transfer time of video-based data through network communication demotivates some users. Also, the present data transfer scheme from mobile devices via network lacks delegation and integrity check of the data. This study focuses on secure multimedia data transfer via network communication from resource constrained devices such as mobile phones and provides a new scheme.

This study focuses on secure multimedia data transfer via network communication from resource constrained devices such as mobile phones and provides a new scheme. This study addresses the following issues of existing techniques that are found in related work: (a) Reduce the communication overhead caused by the 3-way handshake, (b) Find and use a protocol that can help in faster multimedia data transfer and yet ensures network reliability and packet loss, (c) Real-time multimedia data-backup engine for resource-limited devices, (d) Compensate for the packet (i.e., data) loss by introducing a log for both the client and server sides, (e) Establish a mechanism allowing users to share their data and provides advanced access control mechanisms such as delegation, and (f) Integrity of the stored data and access control for each file.

A 2-way handshake. A 3-way handshake of TCP protocol helps improve the data communication's reliability at the cost of extra overhead [8]. In this study, we modified the RTP protocol to allow a 2-way handshake (See Fig 2). Although this modification does not seem to improve communication significantly for smaller files, it improves the data transfer rate for a large data file transfer, such as stream-based multimedia data transfer, as shown in a comparison in Fig. 6. This idea was first implemented by Amiruzzaman and Kim [8], and still seems to be effective. However, one drawback of the 2-way handshake is that it compromises communication reliability. This study already compensated for this using the *log*, which is explained in section 3.1.

Faster multimedia data transfer. UDP protocol is good for multimedia data transfer. However, the UDP protocol is not a good choice for reliable communication. Moreover, this study wants to provide a real-time solution. Therefore, considering the need and quality of communication, a good choice would be the RTP protocol. RTP protocol is often used in multimedia data transmission [11]. This study adopts this protocol and further modifies it to improve performance. The modification part of the RTP protocol is explained in sections 3.1 and 3.1.

A comparison of the TCP and RTP protocol. Suppose a sender sending a packet in T_1 time which is receiving the receiver in T_2 time, again the receiver

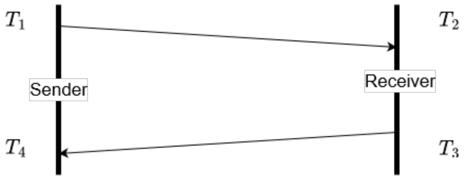


Figure 3: Round Trip Time (RTT) with transmission delay

is sending an acknowledgment in T_3 time which is receiving by the sender end in T_4 time, so the Round Trip Time (RTT) denoted by R will be (see Fig. 3):

$$R = [(T_4 - T_3) + (T_2 - T_1)] \tag{1}$$

The throughput of TCP protocol (TR_t) is,

$$TR_t = \left(\frac{S}{R\sqrt{\frac{2bp}{3} + t(3\sqrt{\frac{3bp}{8}p})(1 + 32p^2)}} \right) = \left(\frac{S}{R\sqrt{p}} \right) \tag{2}$$

here, S is the packet size in bytes, R is a round trip in seconds, t is transmission timeout, p is the average loss rate and b is the number of packets acknowledged. Whereas, RTP protocol the average packet loss p is almost negligible, so we obtain the throughput value for RTP protocol as $TR_r = \frac{S}{R}$.

The transmission rate (TMR) can be calculated as,

$$TMR = \frac{\text{Data-of-a-cycle}}{\text{Duration-of-a-cycle}} = \frac{\text{Packet-size}}{\text{Duration-of-a-cycle}} \tag{3}$$

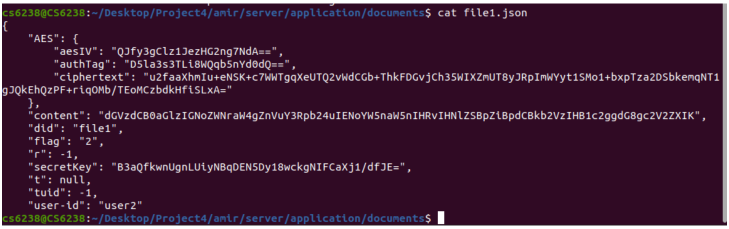
So, the transmission rate for the TCP protocol TMR_t can be denoted as,

$$TMR_t = \frac{\text{Maximal-packet-size} \times \text{Window-size}}{\text{Duration-of-a-cycle}} \tag{4}$$

and, the transmission rate for the RTP protocol TMR_r can be denoted as,

$$TMR_r = \frac{\text{Packet-size} \times \text{Window-size}}{\text{Duration-of-a-cycle}} \tag{5}$$

While RTP packet size is always greater than TCP packet size, hence $TMR_r \geq TMR_t$, and comparing throughput of TCP (TR_t) and throughput of RTP (TR_r), we get $TR_r(\frac{S}{R}) \geq TR_t(\frac{S}{R\sqrt{p}})$



```
cs6238@CS6238:~/Desktop/Project4/amir/server/application/documents$ cat file1.json
{
    "AES": {
        "aesIV": "QJfy3gClz1JezHG2ng7NdA==",
        "authTag": "D5la3s3TLi8WQqb5nYd0dQ==",
        "ciphertext": "u2faaXhmIu+eNSK+c7WWTgqXeUTQ2vWdCGb+ThkFDGvjCh35WIXZmUT8yJRpImWYyt1SMo1+bxpTza2DSbkemqNT1
gJQkEhQzPF+riqOMb/TEoMCzbdkHfiSLxA="
    },
    "content": "dGVzdCB0aGlzIGNoZWNraW4gZnVuY3Rpb24uIENoYW5naW5nIHRvIHNlZSBpZiBpdCBkb2VzIHB1c2ggdG8gc2V2ZXIK",
    "did": "file1",
    "flag": "2",
    "r": -1,
    "secretKey": "B3aQfkwnUgnLUiyNBqDEN5Dy18wckgNIFCaXj1/dfJE=",
    "t": null,
    "tuid": -1,
    "user-id": "user2"
cs6238@CS6238:~/Desktop/Project4/amir/server/application/documents$ 
```

Figure 4: User and file related meta-data to allow access control

Real-time multimedia data backup network-engine. By definition, resource constrained devices lack CPU power, memory, and storage [3]. Therefore, storing and retrieving large video files on resource constrained devices can be challenging. To overcome this problem, this study implements a server-based solution. The server stores and provides video files to resource constrained device users on request. Users can transfer their multimedia data, such as video files and images. As for the communication protocol, this study implements a modified RTP protocol. This protocol is ideal for real-time multimedia data communication. In addition, to improve the data transfer rate real-time slots are used (see Fig. 5). In general, some slots remain idle in real-time communication, marked using orange color in Fig. 5. Those slots are used explicitly in this study to transfer *log* files and *meta-data* information to better utilize the bandwidth and improve the data transfer rate, reliability of the communication, and security.

Packet (i.e., data) loss control. In this study, we used *log* files to control packet loss; the *log* file contains user information along with the chunk/packet number so that both parties can sync and verify if any chunk/packet is missing. This helped to establish reliability and generate re-transmission requests in case of missing chunks/packets. Further, a JSON-based *meta-data* file was stored for user-related information, such as user-id for the file owner, permission information, etc. We used the JSON library to read and write the JSON data (see Fig. 4). Again, a base64-based encoding and `urandom` function was used from Python's os library to generate the unique session token for users.

Data sharing and delegation. The mutual authentication for the secure shared store or 3S model uses the Nginx server [9]. The server used a trusted certificate authority (CA) folder where it had stored the certificate to prove its authenticity. The clients (e.g., client1 and client2) used their private keys to authenticate to the server. Refer to Fig. 1 for architectural design details.

The CA consists of a certificate, a key, and an srl file. The CA.crt file is the certificate file, the corresponding key file is CA.key, and the CA.srl file keeps track of the following available serial number. When we initially generated the client certificates, we used CA.crt to sign the clients' .crt files. This ensured that clients were authentic, and as for the login, public key and private key models were applied. Users sign their message using their private key and send it to the server. While the server verifies the signed message using the users' public key stored on the server (Refer to Fig. 1).

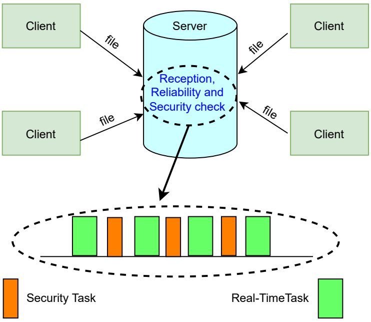


Figure 5: Real-time slots used to improve the data transfer rate

Integrity and access control To ensure the integrity of the data, a MAC is generated for each file transferred to the server. When the server receives a complete file, it generates a MAC for that file and stores it along with its meta-data information. The MAC ensures the integrity of the data, a commonly used technique that helps identify if data is altered. However, to our knowledge, this technique is not used in mobile-based data transfer. Therefore, this study included the MAC to strengthen the security aspects of the stored and transferred data.

4 Results and Discussion

We have implemented a 2-way communication mechanism in RTP protocol and evaluated its performance (For a detailed discussion, refer to Section 3.1). Our results indicate promises and a path to move forward with this proposal. We also simulated a log-based reliability mechanism that helped to determine any missing file and network packets (see Fig. 5).

As shown in Fig. 1, in this proposed study, the client(s) were able to send their certificates and synchronize (SYN) requests with the server. As the certificate from the client is verified by the server, the server sends SYN and acknowledgment (ACK) to the client. After getting the SYN and ACK from the server, the client(s) starts to send the data or request data from the servers. All these are simulated and tested using multiple clients.

In this study, we first recorded and transferred data time during the simulation using TCP protocol. Second, we recorded the data transfer time using

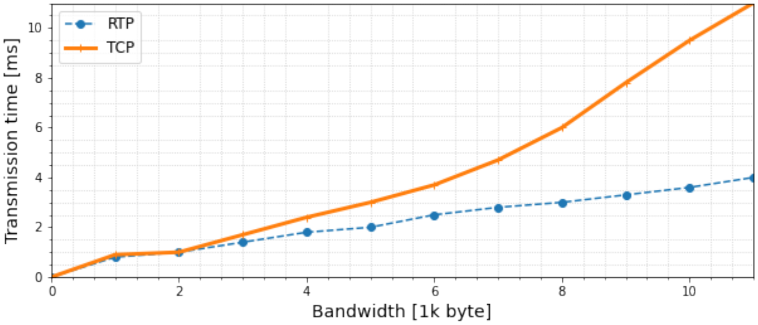


Figure 6: Packet transmission performance of RTP and TCP protocol.

the modified RTP protocol. Third, we compared the data transfer times to understand if the proposed model showed performance changes. Further, we tested the authentication and delegation of the data by sharing it with other users. This process also helped us check the authentication and integration of the data. The simulation environment indicated successful communication and ensured the security aspects of the proposed model.

5 Conclusion

In this study, we tackled the challenges associated with multimedia stream data transfer between a server computer and resource constrained devices. We developed a real-time multimedia data backup network engine for resource constrained devices. We have improved the throughput by introducing an RTP protocol and a two-way handshake for data communication. Also, we compensated for the data loss by introducing a log for both the client and server sides. While the proposed solution shows much promise and solves the challenges found in the related work, a mass implementation and more sophisticated tests are needed, which can be done in future studies.

References

[1] Banu Y Ekren and Anıl Akpunar. “An open queuing network-based tool for performance estimations in a shuttle-based storage and retrieval system”. In: *Applied Mathematical Modelling* 89 (2021), pp. 1678–1695.

[2] Yongkai Fan et al. “Robust End Hopping for Secure Satellite Communication in Moving Target Defense”. In: *IEEE Internet of Things Journal* 9.18 (2022), pp. 16908–16916. DOI: 10.1109/JIOT.2022.3144971.

[3] Anshita Gupta et al. "FedCare: Federated Learning for Resource-Constrained Healthcare Devices in IoMT System". In: *IEEE Transactions on Computational Social Systems* (2023).

[4] Irina Kegishyan. *Mobile Video Statistics.* `https://www.yansmedia.com/blog/mobile-video-statistics`. [Online; accessed 12-March-2023]. 2022.

[5] Selvaraj Kesavan et al. "An investigation on adaptive HTTP media streaming Quality-of-Experience (QoE) and agility using cloud media services". In: *International Journal of Computers and Applications* 43.5 (2021), pp. 431–444.

[6] P Ravi Kumar, P Herbert Raj, and P Jelciana. "Exploring data security issues and solutions in cloud computing". In: *Procedia Computer Science* 125 (2018), pp. 691–697.

[7] Alexander Kunst. *Most used devices for digital videos in the U.S. in 2022.* `https://www.statista.com/forecasts/997109/digital-video-usage-by-devices-in-the-us`. [Online; accessed 12-March-2023]. 2022.

[8] Amiruzzaman Md and Hyoung-Joong Kim. "An Embedded Multi-Agent Based Healthcare Service with Two-way Handshaking Mode". In: *Journal of the Institute of Electronics Engineers of Korea CI* 45.5 (2008), pp. 155–161.

[9] Chris Porter, Sharjeel Khan, and Santosh Pande. "Decker: Attack Surface Reduction via On-Demand Code Mapping". In: *Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2.* 2023, pp. 192–206.

[10] Siqi Ren et al. "Selection-based resampling ensemble algorithm for nonstationary imbalanced stream data learning". In: *Knowledge-Based Systems* 163 (2019), pp. 705–722.

[11] Omar Said et al. "IoT-RTP and IoT-RTCP: Adaptive protocols for multimedia transmission over internet of things environments". In: *IEEE access* 5 (2017), pp. 16757–16773.

[12] Shreyas Sen, Shovan Maity, and Debayan Das. "The body is the network: To safeguard sensitive data, turn flesh and tissue into a secure wireless channel". In: *IEEE Spectrum* 57.12 (2020), pp. 44–49.

An Effective Approach to Closing the Breach Detection Gap (BDG)*

Sydney Raymond[1], David Truong[2], Khalil Davis[2],
Jerry Godwin Diabor[3], David Anyiwo[3]
[1]Welsh School of Foreign Services
Georgetown University, Washington, DC 20057
scr72@georgetown.edu
[2]Department of Computer Science
Valdosta State University, Valdosta, GA 31698
{dqtruong, khdavis}@valdosta.edu
[3]Department of Computer Science
Bowie State University, Bowie, MD 20715
{jdiabor, danyiwo}@bowiestate.edu

Abstract

The goal of this research is to find ways to shorten detection timeframes, which will restrict the amount of time that personally identifiable information (PII) from leaked data is exposed to the public and can be used in cybercrime schemes like phishing. These cyberthreats and hacks have cost many of their victims' customers and ruined their reputations. According to [4], espionage and monetary gain are the two main reasons why breaches of information are carried out. Depending on their capacity to grant access to new data sources or unlock other accounts, certain pieces of data may increase in value on the internet black market. This study will act as a ground-breaking method of providing vital resources and hope to address the difficulties and breach detection gaps that present themselves in identifying them.

1 Introduction

In "Data Breaches," [4] wrote that over the fifteen years prior to the article's publication, over ten billion records had been breached as the result of over 9,000 data breaches in the United States. The author defines a data breach as "an unintentional release of secure or personally identifiable information to an unsecure environment." On an individual level, this can include email addresses, passwords, credit card numbers, and financial information. On a corporate level, it can include business-sensitive trade secrets. Many victims of these cyberattacks and threats have been left with damaged reputations and lost customers. Diedrich [4] asserts that the two most common motivations for conducting data breaches are financial gain and espionage. Pieces of data can gain value on the online black market depending on their ability to provide access to new sources of data or unlock other accounts[9]. The author states that "state-affiliated groups and countries are behind ninety-six percent of espionage motivated breaches." There is a considerable cyber-espionage threat from both Chinese and Russian state-backed hacking groups, according to [5][10]. Furthermore, in this era of the Internet of Things (IoT), the risks and vulnerabilities of network systems are increasing exponentially, as there are over 500,000 new internet users and over 2,000 cyberattacks per day [8, 11]. However, despite the massive security risks associated with data breaches, breaches are often detected by third party groups rather than breach victims, and the average breach detection gap (BDG), the period between a cyberattack and the system's response, exceeds three months [2]. [2] additionally identified multiple instances in which personally identifiable information (PII) remained publicly available for nearly two years. The authors of the paper "Pro-Active Data Breach Detection: Examining Accuracy and Applicability to Personal Information Detected" argued that a reduction in the BDG would reduce the opportunity for cybercrime, indicating the importance of research into gap reduction [2]. Nevertheless, conventional solutions for reducing the BDG have proven to be ineffective and inefficient against emerging cyberattacks and threats[8][7]. Therefore, the aim of this literature review is to conduct a systematic analysis of past and current research studies concerning BDG minimization in order to find an efficient approach to shortening the gap. The systematic analysis of these research studies consisted of categorizing the studies into three foci, studies concerning the identification of factors that contribute to BDG expansion, studies involving potential solutions for closing the BDG, and studies relating to case studies. This specific grouping of literature allows us to identify and construct the proper research concept and methodology for our contribution to the field of information system security.

2 Methodology (Breach Detection Gap (GAP))

There are various methodologies adopted by Breach Detection Gap to ascertain the changes on the networks as they monitor. Any change or gap identified may be an external or misuse by employees or personnel. The method adopted here is hybrid-detection methodology which shall provide better breach detection gap. The model architecture comes with 4 functional blocks, activity blocks that harness events from the monitored environment and analyze other blocks. The hybrid detection is composed of network intrusion detection method and anomaly detection method. Anomaly traffic detection relies on entropy of network attributes and Support Vector Machine (SVM).

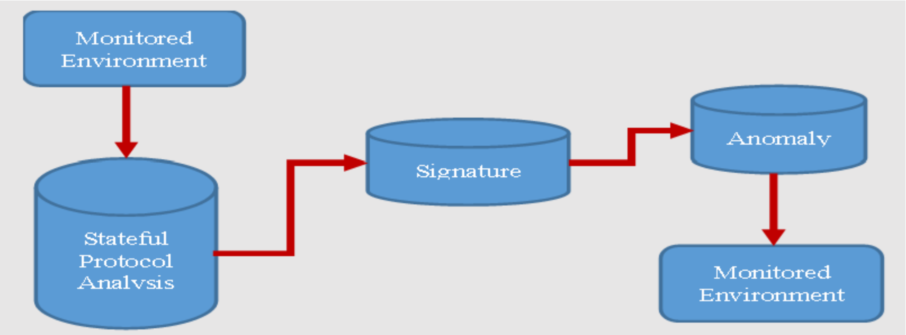


Figure 1: Hybrid Based Methodology Architecture

The hybrid method provided the means of understanding the factors that come into play to create the BDG within the environment of the organization. The table below shows the ways in which the BDG was assessed by combining any one of the factors/types.

Table 1: Breached Detection Gap Factors

Type	Anomaly	Protocol	Signature	Hybrid
Technology	Medium	Low	Low	Low
Human Error	High	High	High	Low
Governmental	Medium	High	High	Medium

2.1 Technology Flaws

The technological flaws in the network system of the organization pose threats coming from hybrid, protocol, and signature. The occurrence can span months, causing damage that puts the organization at risk. The anomaly attempt juxtaposed with the technological flaw shows a moderate level of resistance. However, hybrid technology offers a very low level of resistance to the situation, and there is the possibility of penetration into the networking system. Technological failures include system glitches such as "application failures, inadvertent data dumps, and logic errors in data transfer"[4].

2.2 Human Error

Human error was accounted for as one of the major threat elements for BDG in the organization. The method shows a high human error rate; anomalies, protocols, and signatures give an idea of high-level contributing factors due to human error in the BDG factors[6]. Even though there is a policy in place that frowns on human interventions that create such errors in the system, the policies offer the best resistance interventions that can be applied in all situations to determine the time the error occurred and get notified as soon as possible. However, the implementation has been another matter of concern to consider in certain situations.

2.3 Governmental Policy

The management policies concerning the policies are the communication strategies, requirements, management structures, and response frameworks established by companies or organizations and state or federal governments, respectively[3][1]. The technology policy identified for anomaly and hybrid prevention is not standard and falls under the state's policy concerning technology usage. However, signature and protocol detection policies are in place to provide security measures for unexpected happenings, but implementation is largely the issue in the organization identified.

3 Survey Results

In this section, based on the BDG discussed earlier, the technique to determine the time gap is the running of network tests through behavior-based detection techniques, as this could effectively unearth attacks on the network. A good practice is to use proactive detection survey results, measures of information sources, and network testing to collect inputs via the content approval workflow.

3.1 Human Detection Technique

The three primary factors selected as potential application areas for BDG reduction methods were technology, human error, and corporate and government policy. We selected these as likely BDG factors based on their repeated occurrence in several previous studies as breach causation factors and their possible impact on detection time. Unlike factors like vulnerability to phishing attacks, which only impact whether or not a breach occurs, we believed that these factors could also affect the progression of a data breach, i.e. the BDG.

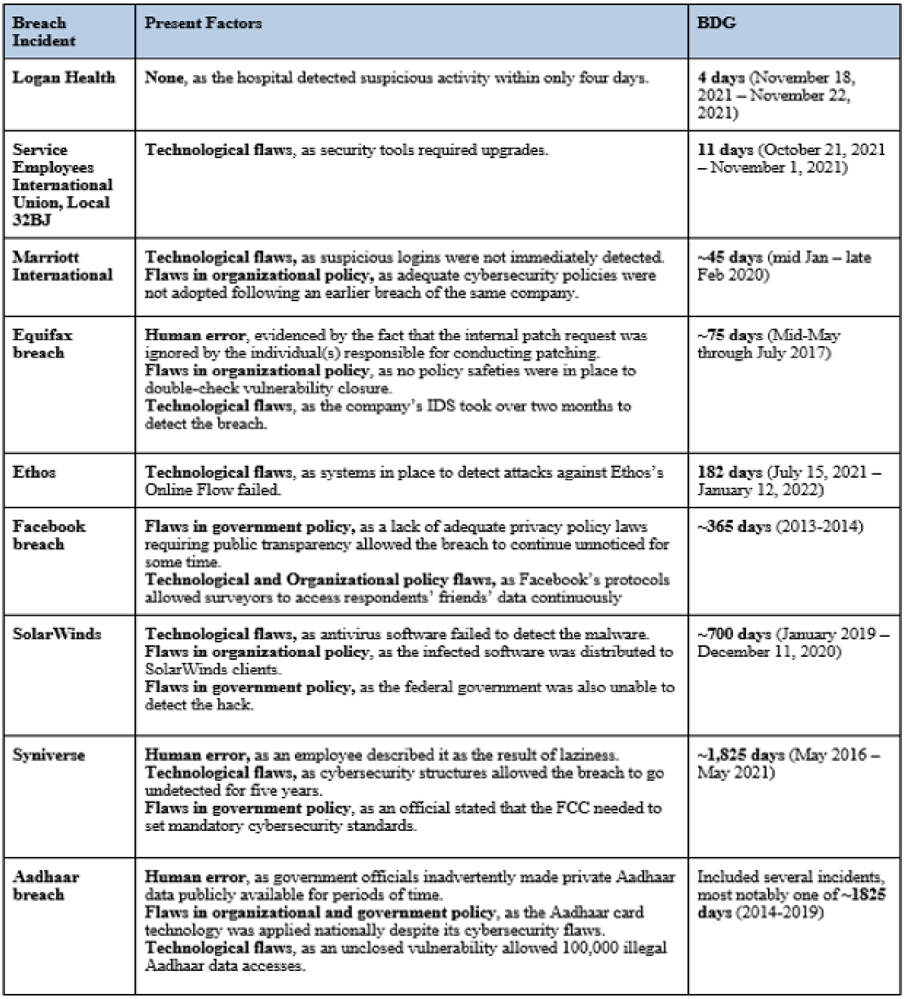
3.2 Graphic Representations of Results

Table 2: Bridge Detection Gap (BDG) Factors

Factors	Response	BDG Factors
Human Error	65.65	17.01
Gov. Policy	95.87	24.84
Corp. Policy	72.75	18.85
Technology Flaws	135.85	35.19
None	15.88	4.11
Total	386	100

3.3 3.2 Model Design, Development and Result

The program created in this study was developed as a multi-tool framework that addresses these factors and the occurrence of a fault in a network system due to a breach. The occurrence of a fault could be catastrophic in certain systems and conditions, and it can be very expensive to fix the fault. Sometimes it could be almost impossible to manually rectify the fault, and single faults are much more likely to occur compared to multiple failures. Containing single failures is more important these days because of improved system reliability. To make sure that non-faulty processes stay mostly unaffected by such local faults and thereby allow for a quicker detection and stabilization, our program permits only a small section of the network around the faulty node to make state changes (See Figure 3). Through extensive experiments, the program has demonstrated the ability to reduce the breach detection time with high accuracy; thus, eliminating errors resulting from false positives. In figure 6, the average stabilization time as the node number increases is depicted, and it seems to depict a linear correlation between stabilization time and average node number. The program addresses the link between technology and the



Breach Incident	Present Factors	BDG
Logan Health	**None**, as the hospital detected suspicious activity within only four days.	**4 days** (November 18, 2021 – November 22, 2021)
Service Employees International Union, Local 32BJ	**Technological flaws**, as security tools required upgrades.	**11 days** (October 21, 2021 – November 1, 2021)
Marriott International	**Technological flaws,** as suspicious logins were not immediately detected. **Flaws in organizational policy,** as adequate cybersecurity policies were not adopted following an earlier breach of the same company.	**~45 days** (mid Jan – late Feb 2020)
Equifax breach	**Human error**, evidenced by the fact that the internal patch request was ignored by the individual(s) responsible for conducting patching. **Flaws in organizational policy**, as no policy safeties were in place to double-check vulnerability closure. **Technological flaws**, as the company's IDS took over two months to detect the breach.	**~75 days** (Mid-May through July 2017)
Ethos	**Technological flaws**, as systems in place to detect attacks against Ethos's Online Flow failed.	**182 days** (July 15, 2021 – January 12, 2022)
Facebook breach	**Flaws in government policy,** as a lack of adequate privacy policy laws requiring public transparency allowed the breach to continue unnoticed for some time. **Technological and Organizational policy flaws,** as Facebook's protocols allowed surveyors to access respondents' friends' data continuously	**~365 days** (2013-2014)
SolarWinds	**Technological flaws,** as antivirus software failed to detect the malware. **Flaws in organizational policy**, as the infected software was distributed to SolarWinds clients. **Flaws in government policy,** as the federal government was also unable to detect the hack.	**~700 days** (January 2019 – December 11, 2020)
Syniverse	**Human error,** as an employee described it as the result of laziness. **Technological flaws,** as cybersecurity structures allowed the breach to go undetected for five years. **Flaws in government policy**, as an official stated that the FCC needed to set mandatory cybersecurity standards.	**~1,825 days** (May 2016 – May 2021)
Aadhaar breach	**Human error**, as government officials inadvertently made private Aadhaar data publicly available for periods of time. **Flaws in organizational and government policy**, as the Aadhaar card technology was applied nationally despite its cybersecurity flaws. **Technological flaws**, as an unclosed vulnerability allowed 100,000 illegal Aadhaar data accesses.	Included several incidents, most notably one of **~1825 days** (2014-2019)

Figure 2: Breached Detection from Companies Time Factors

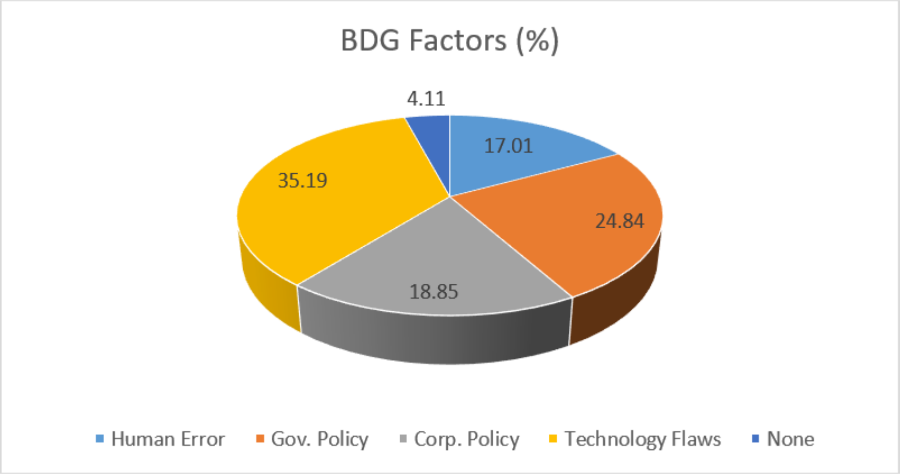


Figure 3: BDG Factors Responses

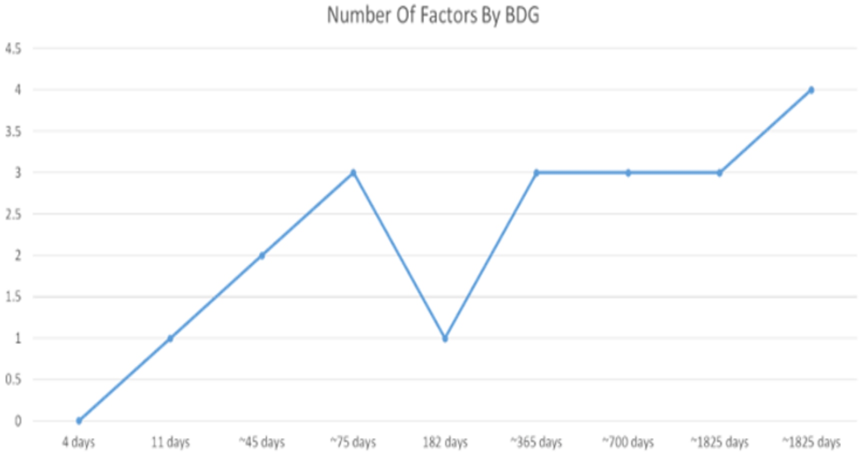


Figure 4: Relationship amidst BDG Time in days Factors

BDG by producing an applicable fault containment, which is a critical feature of stabilizing breached systems, that allows organizations to be able to detect corrupted nodes under a shorter time and to stabilize the network under a shorter period. This is shown in the diagram below:

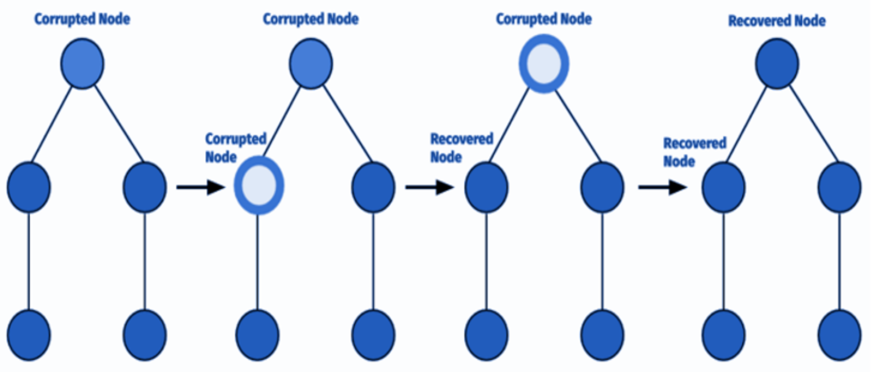


Figure 5: Program Development Concept Architecture

3.4 Algorithm and Java Source Codes

3.4.1 Algorithm

Listing 1: Algorithm

```
// Sample Algorithm for BDG
Attack Structure classification algorithm
Input: Detection and reporting attack structures
Output: Malware attack category
1. if FGd3mmd=FGHmmd=FGlocalgen=MJDmmd=ABSmmd
        =MJlocalgen=nothen
2. malware <-Node-1
3. else
4.      if delShCpy=overFile=no then
5.      malware <- Node-2
6. else
7.      if FGd3mmd = FGHmmd = MJlocalgensym = yes then
8.      malware <- Node-5
9. else
10.      if FGd3mmd = FGHmmdsym = MJlocalgensym = yes then
11.     malware <- Node-3
12. else
```

```
     malware <- Node-4
                       end if
               end if
        end if
    end if = 0
```

3.4.2 Java Codes

Listing 2: Java

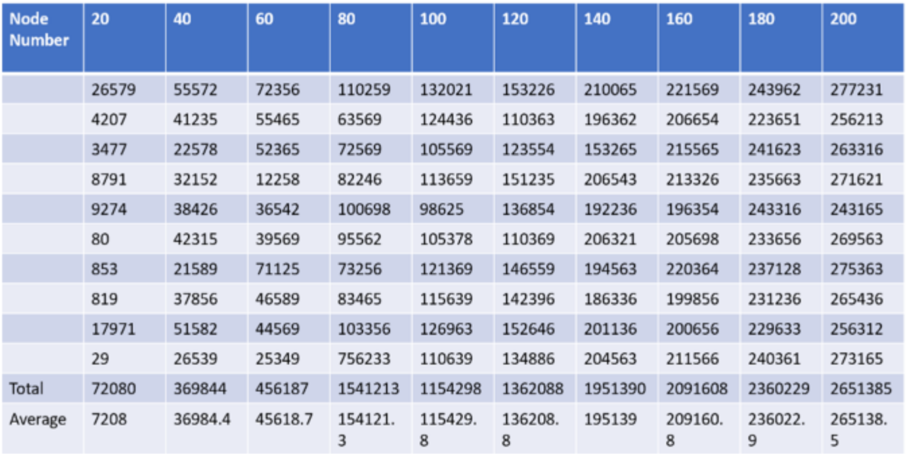
```
// Java Codes for Bridge Detection Gap [BDG]
import java.util.ArrayList;
import java.util.Collections;
import java.util.Random;
import java.util.Scanner;

/*
 * David Tan Truong
 * Raymond Sydney
 * Khalil Davis
 * Jerry Godwin Diabor (PhD-candidate)
 * Dr. David Anyiwo
 * @author
 */
public class Graph {
    public static void main(String[] args) {
        ArrayList<Integer> x_values = new ArrayList<>();
        int size = 21;
        for (int i = 0; i < size; i++) {
            x_values.add(i);
        }
        Collections.shuffle(x_values);
        ArrayList<Node> nodes = new ArrayList<>();
        // generate 20 nodes
        for (int i = 0; i < size; i++) {
            nodes.add(new Node(x_values.get(i)));
        }
        // link them randomly
        linking(nodes);
        long start = System.currentTimeMillis();
        String firstTime =
        java.time.LocalTime.now().toString();
        Node finalNode = processing(nodes);
        System.out.println("Time␣start:␣" + firstTime);
        System.out.println("Time␣end:␣"
        + java.time.LocalTime.now());
        System.out.println("Total␣time␣taken␣for␣execution:␣"
                + (System.currentTimeMillis() - start)
                + "␣milli␣second");
        System.out.println("node␣" + finalNode.getNumber()
        + "␣was␣selected.");
        System.out.println(finalNode.connection());
    }
    private static Node processing(ArrayList<Node> nodes) {
        Random rand = new Random();
        String out = "";
        /*Node t = getRandomNode(rand, nodes);
        t.setState(false);*/
        setFirstNode(nodes);
        ArrayList<String> result = new ArrayList<>();
        System.out.println("First␣List:␣");
        System.out.println("");
        for (Node node : nodes) {
            out += node.toString() + System.lineSeparator();
        }
        System.out.println(out);
        Node finalNode = null;
        do {
            //result.add(out);
            // 2, pick a random node
            out = "";
            Node n = getRandomNode(rand, nodes);
            for (Node node : nodes) {
                out += node.toString() +
                System.lineSeparator();
            }
            // do comparing
            n.comparing(nodes);
            finalNode = n;
        } while (isAllTrue(nodes) == false);
        /*for (String string : result) {
            System.out.println(string +
            System.lineSeparator());
        }*/
        System.out.println("");
        System.out.println("Final␣List␣before␣stopping:");
        System.out.println("");
        System.out.println(out);
        return finalNode;
    }
    private static void setFirstNode(ArrayList<Node> nodes){
        System.out.println("Enter␣a␣number␣from␣0␣to␣"
        + (nodes.size() - 1) + "␣to␣set␣it␣to␣false");
        Scanner scanner = new Scanner(System.in);
        int num;
        do {
            num = scanner.nextInt();
        } while (num < 0 || num >= nodes.size());
        Node t = nodes.get(num);
        t.setState(false);
    }
    public static boolean isAllTrue(ArrayList<Node> nodes) {
        for (Node node : nodes) {
            if (node.getState() == false) {
                return false;
            }
        }
        return true;
    }

    /**
     * linking nodes together
     *
     * @param nodes
     */
    private static void linking(ArrayList<Node> nodes) {
        // one node never connect to itself
        int size = nodes.size() / 4;
        if (size > 5) {
            size = 5;
        }
        Random rand = new Random();
        // for each node
        for (Node node : nodes) {
        // get random number of node to link to this node
        // a node must has 1 connection at least
        int time = (int) (Math.random() * size) + 1;
        // clone the original list
        // so that we could remove item from the cloned one
        // without any worrying
        ArrayList<Node> tmp = (ArrayList<Node>)
        nodes.clone();
        // a node shouldn't link to itself
        tmp.remove(node.getNumber());
        // get reference to list of neighbor of this node
        ArrayList<Node> neighbor = node.getNeighbor();
        for (int i = 0; i < time && tmp.isEmpty() ==
        false; i++) {
        int randomIndex = rand.nextInt(tmp.size());
        Node n = tmp.get(randomIndex);
        neighbor.add(n);
        tmp.remove(randomIndex);
            }
        }
    }
    private static Node getRandomNode(Random rand,
    ArrayList<Node> nodes) {
        int randomIndex = rand.nextInt(nodes.size());
        Node n = nodes.get(randomIndex);
        return n;
    }
}
class Node {
    private int number;
    private int x_value;
    private boolean state;
    private ArrayList<Node> neighbor;
    private static int counter = 0;
    public Node(int x_value) {
        state = true;
        this.x_value = x_value;
        number = counter;
        neighbor = new ArrayList<>();
        counter++;
    }
    public int getNumber() {
        return number;
    }
    public int getX_value() {
        return x_value;
    }
    public boolean getState() {
        return state;
    }
    public ArrayList<Node> getNeighbor() {
        return neighbor;
    }
    public void setState(boolean state) {
        this.state = state;
```

```
}
/**
 * When comparing, if the neighbor x value is smaller
 * than the node then add
 * 1 and keep the default as true if greater than add
 * 10 then flip from what it is currently to the opposite
 * sign. Ultimately what really matter is when the false
 * node flip from false to true and that only happen when
 * that node x value is greater than the rest of it
 * neighbor.
 */
public void comparing(ArrayList<Node> nodes) {
    boolean notFound = true;
    for (Node node : neighbor) {
        if (node.x_value > this.x_value) {
            this.x_value += 10;
            state = !state;
            notFound = false;

            break;
        }
    }
    if (notFound) {
        this.state = true;
        x_value += 1;
    }
}

public String connection() {
    String result = "node␣" + number;
    result += "␣{␣";

    result += neighbor.get(0).number;
    for (int i = 1; i < neighbor.size(); i++) {
        Node get = neighbor.get(i);
        result += ",␣" + get.number;
    }
result += "␣}␣-->␣" + state + System.lineSeparator();
result += "the␣new␣x␣value␣for␣node␣"
      + number + ":␣";
    result += neighbor.get(0).x_value;
    for (int i = 1; i < neighbor.size(); i++) {
        Node get = neighbor.get(i);
        result += ",␣" + get.x_value;
    }
```

```
        result += "␣" + x_value;
        return result;
    }

    /**
     *
     * @return description of the node with its neighbor
     */
    @Override
    public String toString() {
        String result = "Node:␣␣->␣";
        result += "␣{␣";

        result += neighbor.get(0).number;
        for (int i = 1; i < neighbor.size(); i++) {
            Node get = neighbor.get(i);
            result += ",␣" + get.number;
        }

        result += "␣}␣->␣" + number + "␣{␣" + state + "␣}"
                + System.lineSeparator();

        result += "X-value:␣";
        result += neighbor.get(0).x_value;
        for (int i = 1; i < neighbor.size(); i++) {
            Node get = neighbor.get(i);
            result += ",␣" + get.x_value;
        }
        result += "␣" + x_value;

        return result;
    }
}
```

4 Discussion and Conclusion

The research identified three potential BDG factors. Using content analysis of several case studies coupled with hypotheses involving the association between each factor (excluding external factors) and the BDG A system design was developed to address the most significant of these factors and therefore aims to reduce the BDG. The program developed was designed to reduce breach detection time and is highly accurate, eliminating errors resulting from false positives. From the data obtained, it was noted that technology occurred the most frequently among the examined factors in selected case studies (Figure



Node Number	20	40	60	80	100	120	140	160	180	200
	26579	55572	72356	110259	132021	153226	210065	221569	243962	277231
	4207	41235	55465	63569	124436	110363	196362	206654	223651	256213
	3477	22578	52365	72569	105569	123554	153265	215565	241623	263316
	8791	32152	12258	82246	113659	151235	206543	213326	235663	271621
	9274	38426	36542	100698	98625	136854	192236	196354	243316	243165
	80	42315	39569	95562	105378	110369	206321	205698	233656	269563
	853	21589	71125	73256	121369	146559	194563	220364	237128	275363
	819	37856	46589	83465	115639	142396	186336	199856	231236	265436
	17971	51582	44569	103356	126963	152646	201136	200656	229633	256312
	29	26539	25349	756233	110639	134886	204563	211566	240361	273165
Total	72080	369844	456187	1541213	1154298	1362088	1951390	2091608	2360229	2651385
Average	7208	36984.4	45618.7	154121.3	115429.8	136208.8	195139	209160.8	236022.9	265138.5

Figure 6: Time Utilization and Number of Nodes, Addressing BDG Factors

2). Therefore, we focused primarily on solving technological flaws that led to increased breach detection times. This goal led to the development of our breach detection algorithm and code (Figure 5). Additionally, the observation came as a positive correlation between breach detection time and the number of present factors, which can be seen in Figure 2 and Figure 5, although we recognize that a full analysis of this possible correlation requires further study. We additionally noted that the frequency of each factor tended to increase with rising breach detection time (Figure 3). We believe that these results may signify that factors may compound, leading to longer breach detection times, and that each of these factors may indeed directly correlate with longer breach detection times. It can be noted that there was an inconsistency in Figure 3 at 182 days. Other breaches within this time range (2 months to 1 year) tended to show a wider variety of factors than the 182-day Ethos breach, which only involved technological flaws, according to our content analysis. Again, I identified two possible explanations for this. It is possible that there were other factors at play in this breach that were not disclosed to the public, causing us to unintentionally underrepresent the number of factors present. It is additionally possible, however, that flawed technology simply had a larger impact in this breach than in other breach incidents. Upon closer examination of the facts of the breach, it was noticed that the hackers involved in the breach used tools deliberately crafted to circumvent the specific IDS that Ethos had in place for its online flow (Demas). It is believed that these targeted strategies could have resulted in an unusually extended breach detection time.

5 Acknowledgements

The National Science Foundation, Award 1757924, supported this research.

References

[1] Kim Whatt Gary Ang. “A Case Study for Cyber Incident Report in Industrial Control Systems”. PhD thesis. Massachusetts Institute of Technology, 2022.

[2] Johnny Botha, MM Eloff, and Ignus Swart. “Pro-active data breach detection: examining accuracy and applicability on personal information detected”. In: *Proceedings of the 11th International Conference on Cyber Warfare and Security (ICCWS).* Vol. 12. 2016, pp. 47–55.

[3] Long Cheng, Fang Liu, and Danfeng Yao. “Enterprise data breach: causes, challenges, prevention, and future directions”. In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 7.5 (2017), e1211.

[4] Drew Diedrich. “Data Breaches”. In: *Geo. L. Tech. Rev.* 4 (2019), p. 315.

[5] Faisal Jamil et al. “Towards secure fitness framework based on IoT-enabled blockchain network integrated with machine learning algorithms”. In: *Sensors* 21.5 (2021), p. 1640.

[6] Faouzi Kamoun and Mathew Nicho. “Human and organizational factors of healthcare data breaches: The swiss cheese model of data breach causation and prevention”. In: *E-Health and Telemedicine: Concepts, Methodologies, Tools, and Applications.* IGI Global, 2016, pp. 1299–1319.

[7] T Lyles. “Marriott discloses another security breach that may impact over 5 million guests”. In: *The Verge. Retrieved March* 10 (2020), p. 2020.

[8] Joao Amaral Santos, Pedro RM Inacio, and Bruno MC Silva. “Towards the use of blockchain in mobile health services and applications”. In: *Journal of Medical Systems* 45 (2021), pp. 1–10.

[9] T Saranya et al. “Performance analysis of machine learning algorithms in intrusion detection system: A review”. In: *Procedia Computer Science* 171 (2020), pp. 1251–1260.

[10] Ping Wang and Christopher Johnson. “Cybersecurity incident handling: a case study of the Equifax data breach.” In: *Issues in Information Systems* 19.3 (2018).

[11] Ping Wang and Sun-A Park. “Communication in Cybersecurity: A Public Communication Model for Business Data Breach Incident Handling”. In: *Issues in Information Systems* 18.2 (2017).

Industry Trends in Software Engineering: Alumni Perspectives*

Karen Anewalt[1] and Jennifer Polack[2]
Department of Computer Science
University of Mary Washington, Fredericksburg, VA 22401
anewalt@umw.edu[1], jenniferpolack@gmail.com[2]

Abstract

It is important for computer science curricula to prepare graduates for their future careers. Alignment efforts between academia and industry benefit both communities. Having data about current industry trends, including tools and critical experiences, allows academia to adjust course assignments and curricula to provide relevant and needed material in today's computer science job market. We present survey responses from industry professionals related to tool, project, communication, and collaboration experiences essential for new employees. The collected data can be used to update and enhance current assignments across curricula. Responding to industry trends and demands can give future computer science professionals valuable experience as they begin their careers.

1 Introduction and Background

Alignment efforts between industry and academic environments bring various benefits to both communities. Recent research suggests that it may be easier to align tools and techniques currently used in industry to academic courses rather than wait for academia to influence industry best practices, because industry environments are sometimes bound to use particular tools and languages originating from project requirements or company policies [6]. In order to inform curricula adjustments that balance the theoretical and practical content of academic programs and to prepare graduates for professional expectations, academia frequently seeks feedback from industry partners and alumni in the computer science field.

Much research has investigated connections between academia and real-world trends. Recent studies [5, 2], showed that academic project-based learning and customer-driven project experiences are important for preparing industry professionals. Heggen and Cody [3] reported that student engagement in real-world projects result in increased student confidence in their software engineering skills and soft skills. Communication skills are consistently identified as the mostly highly-rated professional skill needed by graduates. Akdur [1] confirms that this remains true, but does not indicate the types of oral and written communication experience that best prepare students for industry.

Surveys are useful for collecting feedback from both students and industry. One recent survey [4] was conducted seeking an understanding of student perceptions of how undergraduate capstone software projects complement traditional classwork by providing real-world software engineering exposure. The survey was conducted over six years and included a qualitative analysis of 2,203 quotes collected from 167 students from 18 universities over six academic terms. The authors concluded students valued performing real tasks on real projects under the guidance of real mentors.

To inform updates to our own program at the University of Mary Washington, we developed a new survey of eleven questions specifically related to professional experiences as software engineers and developers. The goal of the survey was to determine the tools and techniques currently used by industry and also identify project experiences that would prove useful to new computer science professionals. Since the computer science field is constantly changing, CS programs must keep up with industry expectations. Particularly since industry experiences and culture have shifted recently, as more people are working remotely, current data has the potential to identify gaps between current academic experiences and expectations for professional practice. Academics have the opportunity to re-align academic software engineering courses to better prepare graduates for experiences relevant to current industry practices.

2 Industry Survey Methodology

A list of alumni emails was obtained from the Office of Alumni Relations.The survey was sent to 1057 alumni from the University of Mary Washington who graduated with a computer science degree from the 1990s through today. The University of Mary Washington is a small liberal arts college where the Department of Computer Science currently supports two majors: Computer Science and Cybersecurity. However, in the period from 1990, majors in Computer Science, Computer Information Systems, Geographic Information Systems, and Cybersecurity have been offered at various times. The majority of alumni surveyed carried specializations in traditional computer science, but other specializations were also included.

The survey addresses the following topics.

- Year they graduated,
- Percentage of time engaged in remote work,
- Project management tools commonly used,
- Documentation collaboration tools commonly used,
- Communication tools commonly used,
- Resources used for troubleshooting,
- Professional document experiences that are essential for graduating undergraduates,
- Professional presentation experiences that are essential for graduating undergraduates,
- Use of pair programming,
- Types of projects and clients that are important in student project experiences,
- And their most valuable experience during their degree.

3 Survey Data Analysis

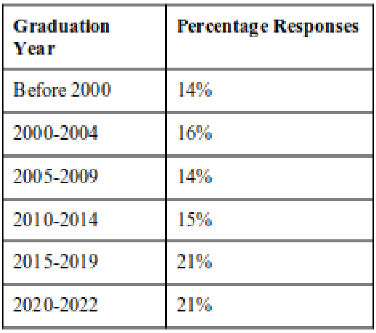
We received 186 responses to the survey and the response group ranged from alumni that graduated from the 1990s through 2022. The responses were fairly evenly distributed among each group of five years as shown in Table 1.

Of the 186 respondents, 180 of them report that they are currently employed in the computer science field. Because we are focused on professional experiences in the computing field, we focus our analysis on these 180 responses.

In order to determine the prevalence of remote telework among the respondents, we asked "On average over the past calendar year, what percentage of your professional work in the computer science/tech field has been remote?" Table 2 shows that more than half of the respondents indicated that they worked remotely more than 60% of the time and more than 80% of them responded that they work remotely more than 20% of the time, so at least one day per week. This is a drastic change from previous decades where telework was much less common.

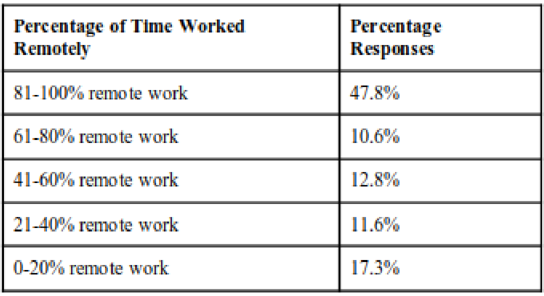
With telework becoming increasingly common over the past several years, we hypothesized that professionals may be using more tools to facilitate remote

Table 1: Percentage of Respondents from Graduation Years



Graduation Year	Percentage Responses
Before 2000	14%
2000-2004	16%
2005-2009	14%
2010-2014	15%
2015-2019	21%
2020-2022	21%

Table 2: Frequency of Remote Work Among Respondents



Percentage of Time Worked Remotely	Percentage Responses
81-100% remote work	47.8%
61-80% remote work	10.6%
41-60% remote work	12.8%
21-40% remote work	11.6%
0-20% remote work	17.3%

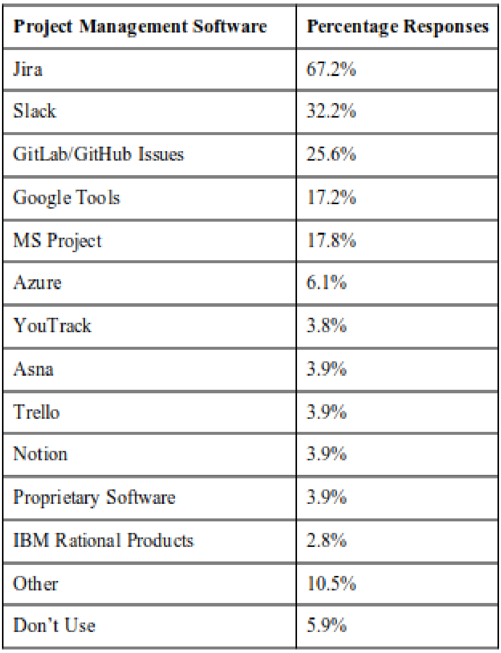
group collaboration specifically related to project management, document collaboration, and team communication. We asked several questions to identify trends.

- Professionally, which project management software do you use frequently? Select up to 3.
- Professionally, what document collaboration tool do you use most frequently?
- Professionally, what communication tool do you use frequently for team communication? Select up to 3.

For each question, we provided a list of popular tools and also allowed respondents to type in additional tools that weren't included. Tables 3, 4, and 5 show the responses. Responses indicate that Jira (67%), Slack (32%), GitHub and GitLab (26%) are popular tools used to facilitate project management tasks. Responses show that Microsoft Teams and Atlassian Confluence are popular choices for document collaboration, with 46% and 31% percent responses respectively. Several tools are widely popular for communication collaboration including Microsoft Teams (71%), email and text (59%), Slack (45%), and Zoom (33%).

One surprising result was the low ranking of Microsoft Project as a project management tool, particularly given the popularity of Microsoft Teams and OneDrive as document collaboration tools and communication tools.

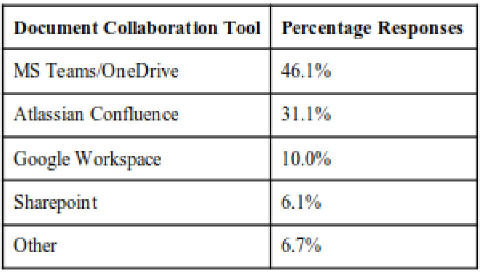
Table 3: Project Management Tools Used Frequently for Collaboration



Project Management Software	Percentage Responses
Jira	67.2%
Slack	32.2%
GitLab/GitHub Issues	25.6%
Google Tools	17.2%
MS Project	17.8%
Azure	6.1%
YouTrack	3.8%
Asna	3.9%
Trello	3.9%
Notion	3.9%
Proprietary Software	3.9%
IBM Rational Products	2.8%
Other	10.5%
Don't Use	5.9%

We were also interested in discovering what tools or platforms industry professionals find the most helpful when troubleshooting issues or seeking support. Of the 180 respondents that said they work in the computing field, 173 said that they are involved in writing code and responded to the question, "What resource do you use most frequently for support or troubleshooting?" Respondents could select their most frequent source of support from a list of

Table 4: Document Collaborations Used Frequently for Collaboration



Document Collaboration Tool	Percentage Responses
MS Teams/OneDrive	46.1%
Atlassian Confluence	31.1%
Google Workspace	10.0%
Sharepoint	6.1%
Other	6.7%

options or provide a different response. As shown in Table 6, the most popular tool cited was StackOverflow, with 52% favoring it. Personal consultation with colleagues was favored by a quarter of the respondents, showing the continuing need for students to develop strong collaboration skills. Assistive AI tools including ChatGPT, not widely available until very recently, received the top selection for a fewl respondents. It will be important for academia to continue to monitor how AI tools influence industry in the coming years.

In order to determine whether pair programming is used widely in industry, we asked, "How frequently do developers use pair programming at your organization?"

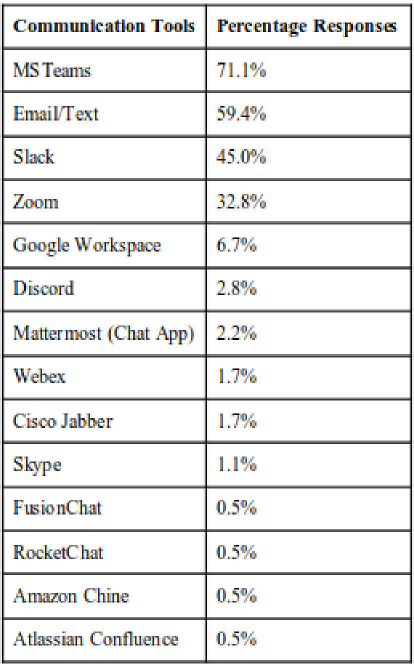
The majority of the respondents, 72%, reported that pair programming is used at least occasionally at their organization. However, pair programming does not appear to be a popular technique for day-to-day software development tasks, as less than 7% of responses say pair programming is used more than half of the time.

We also asked several questions to determine the types of communication experiences industry expects from entry level employees. We asked:

- Which types of professional documents are essential experiences for students prior to graduation?
- Which types of professional presentations are essential experiences for students prior to graduation?
- How would you rate the importance of involving a non-technical client as part of an undergraduate student project experience?

The responses are shown in Table 8, 9, and 10. Respondents indicated that many of these common industry documents and presentation types are essential for undergraduate students. More than half of the respondents said that experience writing requirements documents, test plan documents, and API

Table 5: Communication Tools Used Frequently for Collaboration

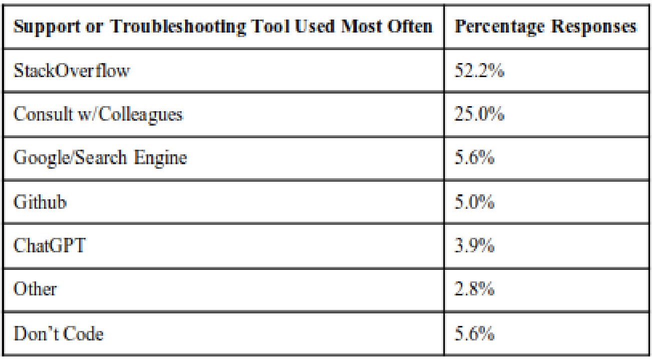


Communication Tools	Percentage Responses
MS Teams	71.1%
Email/Text	59.4%
Slack	45.0%
Zoom	32.8%
Google Workspace	6.7%
Discord	2.8%
Mattermost (Chat App)	2.2%
Webex	1.7%
Cisco Jabber	1.7%
Skype	1.1%
FusionChat	0.5%
RocketChat	0.5%
Amazon Chine	0.5%
Atlassian Confluence	0.5%

documentation are essential. Half or more of the respondents indicate that communication experiences with code reviews, sprint planning, sprint reviews, and project status briefings are essential. In addition to the provided options, several respondents mentioned that students should gain experience in doing project demos for clients. And more than 72% of the respondents indicated that having a real client was essential or a valuable component of a student project experience.

We also asked alumni to identify an experience that was important as they started their career. The question was: “Reflecting back on your time as a student, what experience was the most valuable as you began your career?” The responses, shown in Table 11, indicate that project experiences and internships were the most valuable, with 85% of the responses.

Table 6: Support or Troubleshooting Tool Used Most Often



Support or Troubleshooting Tool Used Most Often	Percentage Responses
StackOverflow	52.2%
Consult w/Colleagues	25.0%
Google/Search Engine	5.6%
Github	5.0%
ChatGPT	3.9%
Other	2.8%
Don't Code	5.6%

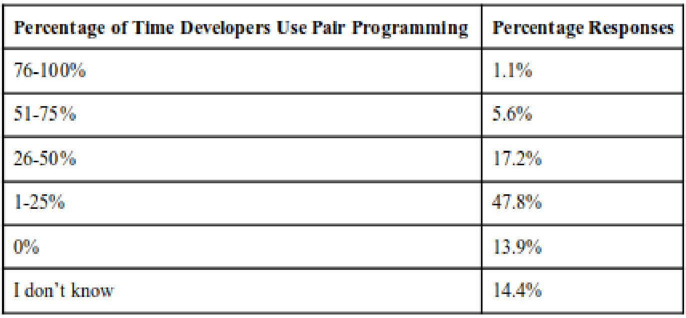
4 Implications for Undergraduate Curriculum

One goal of our survey was to identify industry trends to update tools and assignments in our curriculum to better align with current industry practices. A second goal is to gain insight from alumni about student experiences that they found helpful as a recent graduate (or would value from today's new graduates seeking entry level positions).

The survey responses show that software project experience were impactful, with most respondents saying that project experience in a course, internship, or personal project were the most valuable experiences in their undergraduate education. We conclude that it is essential for computer science curricula to provide a range of project experiences as part of required courses. Of particular importance is the inclusion of long term, collaborative projects that allow students to gain experience in key communication and documentation relevant in the real world including requirements documents, test plans, sprint planning, sprint reviews, and project status updates. These types of long range projects also provide opportunities to involve real world, non-technical clients, which respondents rated as a valuable or essential part of an undergraduate project experience. Large scale projects are also ideal places to expose students to project management tools such as Jira and collaboration tools like MS Teams. And collaborative projects provide opportunities for students to engage in the types of support tools that professionals use including StackOverflow and consulting with peers.

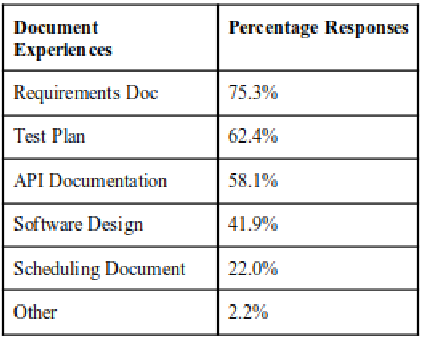
At the University of Mary Washington, we include long term, collaborative projects in several required courses within the curriculum: object-oriented

Table 7: Percentage of Time Developers Use Pair Programming



Percentage of Time Developers Use Pair Programming	Percentage Responses
76-100%	1.1%
51-75%	5.6%
26-50%	17.2%
1-25%	47.8%
0%	13.9%
I don't know	14.4%

Table 8: Document Experiences that are Essential for Students

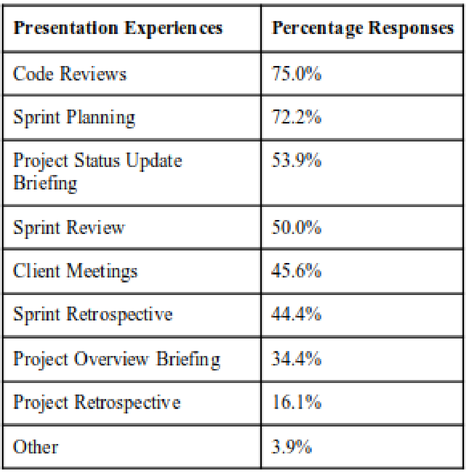


Document Experiences	Percentage Responses
Requirements Doc	75.3%
Test Plan	62.4%
API Documentation	58.1%
Software Design	41.9%
Scheduling Document	22.0%
Other	2.2%

analysis and design (CS2), databases, and software engineering. These courses allow students to practice and develop collaborative skills over time and engage in different aspects of the software lifecycle. CS2 focuses primarily on the planning and implementation phases. The database course focuses primarily on the implementation phase and group communication. And the software engineering course uses Agile to immerse students in the full software lifecycle and provides the opportunity to work with a non-technical, real-world client.

While many of the tools and experiences valued by the survey respondents are already included in our curriculum, especially in the existing software engineering capstone course, we do see opportunities to build on current strengths. For example, because respondents indicated that API documentation is the

Table 9: Presentation Experiences that are Essential for Students



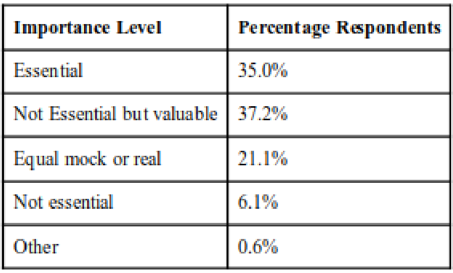
Presentation Experiences	Percentage Responses
Code Reviews	75.0%
Sprint Planning	72.2%
Project Status Update Briefing	53.9%
Sprint Review	50.0%
Client Meetings	45.6%
Sprint Retrospective	44.4%
Project Overview Briefing	34.4%
Project Retrospective	16.1%
Other	3.9%

most essential type of documentation that graduates should have experience with, we are planning to make this a more prominent component of the CS2 course. We are also looking for ways to integrate code reviews more consistently across the curriculum beginning in CS1. We will also be increasing the instructor emphasis on the importance of consulting with team members and peers when working on collaborative projects.

Within our software engineering capstone course, we see additional opportunities to expand on the existing course assignments. In the course, students spend the full semester creating custom software for non-profits and other local organizations through the customization of free and open source software (FOSS) in collaboration with the Non-Profit FOSS Institute (NPFI) organization. Most presentations and documents identified as important by the survey responses are already included in the course. Some of the opportunities to better align the assignments within the software engineering course with industry practices are listed below.

- We are considering ways to encourage different project teams to perform code reviews of other groups' projects.
- The course projects already follow an Agile approach and students engage in both sprint planning and sprint retrospectives. However, in the current format, these communication activities take place within the group and the instructor does not have an opportunity to provide feedback. We

Table 10: How Important is Involving a Non-technical Client in a Student Project Experience



Importance Level	Percentage Respondents
Essential	35.0%
Not Essential but valuable	37.2%
Equal mock or real	21.1%
Not essential	6.1%
Other	0.6%

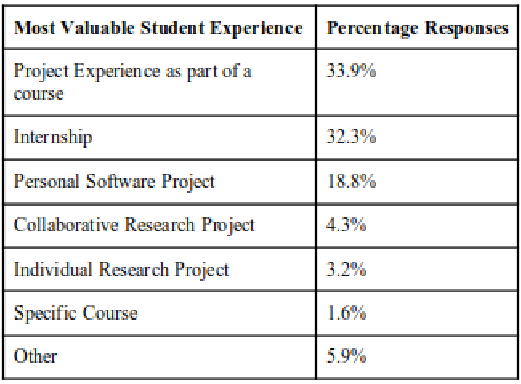
are considering making these tasks an assignment and requiring groups to submit an agenda or detailed meeting report discussing issues considered. This will allow instructors to provide more substantive feedback and oversight on the sprint planning and sprint retrospective process.

- Because many survey respondents indicated that project status updates are important presentation experiences, with many noting that students should be required to practice communicating to a non-technical audience, we are considering adding an assignment requiring each project group to provide a status update to their client (via MS Teams or Zoom) at several points during the semester. A recording of the meeting can be submitted to the course instructor for feedback on their communication strengths and opportunities for improvement.
- We are considering more heavily integrating MS Teams and Atlassian tools because survey responses indicated that they are used heavily by employers. We are also considering using the Professional or Business version of Slack.

5 Conclusion

Many computer science students enter the workforce following their degree. It is important for computer science curricula to prepare graduates for their future careers. Data about current industry trends, including tools and experiences, allows academia to adjust course experiences to provide relevant and needed material in today's computer science job market. We anticipate using the survey responses that we received supporting the importance of project experiences, communication experiences, and collaboration experiences to update and enhance current assignments across our curriculum, but particularly in our

Table 11: Most Valuable Student Experience



Most Valuable Student Experience	Percentage Responses
Project Experience as part of a course	33.9%
Internship	32.3%
Personal Software Project	18.8%
Collaborative Research Project	4.3%
Individual Research Project	3.2%
Specific Course	1.6%
Other	5.9%

software engineering capstone course. Responding to industry trends and demands can give future computer scientist professionals valuable experience as they begin their careers.

References

[1] Deniz Akdur. “Analysis of Software Engineering Skills Gap in the Industry”. In: *ACM Trans. Comput. Educ.* 23.1 (Dec. 2022).

[2] Orges Cico and Letizia Jaccheri. “Industry Trends in Software Engineering Education: A Systematic Mapping Study”. In: *Proceedings of the 41st International Conference on Software Engineering: Companion Proceedings*. ICSE ’19. Montreal, Quebec, Canada: IEEE Press, 2019, pp. 292–293.

[3] Scott Heggen and Cody Myers. “Hiring Millennial Students as Software Engineers A Study in Developing Self-Confidence and Marketable Skills”. In: June 2018.

[4] Reid Holmes, Meghan Allen, and Michelle Craig. “Dimensions of Experientialism for Software Engineering Education”. In: *Proceedings of the 40th International Conference on Software Engineering: Software Engineering Education and Training*. ICSE-SEET ’18. Gothenburg, Sweden: Association for Computing Machinery, 2018, pp. 31–39. ISBN: 9781450356602.

[5] Sami Jantunen and Timo Hynninen. “Increasing Industry-Academia Collaboration: Types of Regional Software Engineering Companies and Their Needs from Academia”. In: *Proceedings of the 2022 European Symposium on Software Engineering*. ESSE ’22. Rome, Italy: Association for Computing Machinery, 2023, pp. 43–47. ISBN: 9781450397308.

[6] Dusica Marijan and Sagar Sen. “Good Practices in Aligning Software Engineering Research and Industry Practice”. In: *SIGSOFT Softw. Eng. Notes* 44.3 (Oct. 2020), pp. 65–67. ISSN: 0163-5948.

The Hunt for Cybersecurity Data: Exploring the Availability of Open Datasets for Cybersecurity Scientific Research[*]

Michelle Liu, Diane Murphy, Nathan Green
School of Technology and Innovation
Marymount University
Arlington, VA 22201
{xliu,dmurphy,ngreen}@marymount.edu

Abstract

The paper looks at how cybersecurity can successfully be transitioned from a practical-focused field to an academic science, albeit a combination of hard sciences such as computer sciences and social sciences, focusing on human behavior. One of the attributes of a science is its basis on a hypothetico-deductive model and the verification and validation of experimental work through repetition. To accomplish this, it is necessary to identify and access cybersecurity datasets, while also recognizing issues such as privacy and data security. The authors look at the science of cybersecurity and what still needs to be done to enable academic researchers to readily identify, access, and use quality datasets. The paper summarizes the issues with current practices and suggests two methods, synthetic data generation and curating a centrally-available list of cybersecurity data sets. The authors' plan for future work in this area is outlined.

1 Introduction

The rapid growth of the digital ecosystem in which we live now brings about a multitude of cybersecurity challenges, including compliance gaps in various sectors [7, 13]; conspicuous discrepancies in skills and talents [4]; and the advanced, persistent threats (APTs) imposed by malicious actors, including nation states [19]. Integration of hardware platforms, software applications with new software development techniques, and new infrastructures such as cloud and quantum computing, have paved the way for the continuing development of technological advancements such as smart homes, intelligent transport systems, and virtual reality, along with increasing attack surfaces and threat vectors to digital assets, including the critical infrastructure [26]. The cybersecurity profession struggles to keep pace with the evolving tactics of cyber attackers, as well as the potential impact of human error or user negligence.

Since the first widely known document on the "Science of Cybersecurity" was published [24], the cybersecurity field has been undergoing a significant transformation. Historically, cybersecurity has been considered a reactive, practical-focused field with an emphasis on developing skills and toolkits to combat existing threats and vulnerabilities. However, there is an increasing recognition that the practice of cybersecurity needs a structured and systematic approach to become proactive and tackle evolving cyber threats and risks [5, 12, 23, 25, 27].

In line with the progression of other fields, such as medicine, cybersecurity is transitioning from a protoscience to a more established and recognized discipline [27]. This evolution is evident, for example, through the rise in the number of Doctor of Science (DSc) and PhD programs specializing in cybersecurity. These academic programs provide opportunities for researchers to contribute to the scientific advancement of the field, foster innovation, and cultivate a deeper understanding of cybersecurity principles and practices.

However, one of the major obstacles that doctoral students are facing is the difficulty in searching and accessing relevant datasets for their research projects. Despite large amounts of security-related data being generated and produced at an unprecedented velocity, most of these datasets are either proprietary or not shared further with the public or have limited accessibility beyond their original scope and purpose [29]. This is seen as an inhibitor to the advancement of cybersecurity as a science because it has been widely recognized that publicly available datasets are invaluable for scientific research. Without being able to easily find and access suitable existing datasets, cybersecurity academic researchers often have to generate their own datasets, which are frequently limited by time and data availability. In addition, doctoral students and faculty are unable to replicate experiments and compare the implementation and performance of tools using existing data. Replication is considered a key factor in the confidence given to a specific piece of scientific research [17].

We believe that open datasets are essential to further cybersecurity as a science, particularly because research and practice in cybersecurity are becoming increasingly data-driven [16]. There are some open datasets available on the Internet, shared by both organizations and individuals, for example IEEE DataPort, GitHub, Kaggle, and the Registry of Open Data on AWS. However, these datasets are not necessarily focused on cybersecurity, and even if some are, they may be archaic, irrelevant to the ever-changing cybersecurity landscape of today, scattered over the Internet, or fragmented in terms of topics, domains, and the nomenclature used to describe them. Therefore, we call for additional efforts to make more easily accessible and well-organized open datasets available for cybersecurity research. We also strive to explore potential methods for organizing or generating datasets in order to promote transparency and strengthen the overall credibility and trustworthiness of the field, thereby substantiating cybersecurity as a science.

The authors seek to identify and evaluate the extent to which open datasets are readily accessible for cybersecurity inquiry as well as to challenge the limitations associated with their usage. Furthermore, they aim to shed light on the implications of open datasets in enhancing the reproducibility and replicability of cybersecurity research. Ultimately, the paper aims to provide insights and recommendations on the utilization and promotion of open datasets in the realm of academic cybersecurity research.

2 Cybersecurity as a Science and Social Science

Compared to those full-fledged disciplines such as computer science or chemistry, the cybersecurity discipline identity is blurred due to overlap and interconnection with other reference disciplines such as parts of computer science, sociology, psychology, economics, electrical engineering, and data science, among others. Over the past decade, public interest has increased in developing a science of cybersecurity [12]. An emerging view now being proposed is to examine cybersecurity through the scientific method and conduct experiments that can be evaluated, replicated, and verified [5].

However, the field of cybersecurity has distinct challenges when it comes to establishing itself as a science. [10] enumerated four key issues, including adaptive adversaries, absence of invariant laws, evolving technology, and the lack of a hypothetico-deductive model. The last challenge is particularly relevant to the context of this work. To be recognized as a science, cybersecurity should adhere to a hypothetico-deductive model, involving the formulation of hypotheses based on observations, the creation of verifiable predictions, and the evaluation of these predictions against new observations. Availability of open datasets for research and associated replication studies are critical for such a model. Replication is a critical challenge in the field of cybersecurity, mirroring the broader landscape of computer science research [3]. Just like other disciplines, cybersecurity research can greatly benefit from adopting steps to enhance replication efforts. By employing more rigorous evidentiary criteria, embracing experimental preregistration so separating hypothesis-generating (exploratory) from hypothesis-testing (confirmatory), and establishing data and artifact repositories, the cybersecurity community can promote greater openness and much needed transparency and reliability in its research practices [3].

There are three main reasons prompting the pursuance of a scientific approach towards cybersecurity. Firstly, better understanding potential cyber threats and adversarial tactics is crucial to effectively countering them [15]. By treating cybersecurity as a science, we can analyze threats, their causes, and potential impacts with more robust and verifiable research and investigative procedures. Secondly, a scientific approach complements practical strategies by providing evidence-based insights and proactive risk mitigation [6]. Additionally, it encourages transparency, replication, standardized frameworks, knowledge sharing, and interdisciplinary collaboration. Treating cybersecurity as a systematic and empirical field of study enhances its credibility, reliability, and the development of consistent educational programs to support the field. In summary, a scientific approach strengthens our defenses, helps us stay ahead of cyber criminals, and helps ensure digital safety and security and national security.

It has been acknowledged that various branches of scientific disciplines have overcome difficulties that were once considered unique and insurmountable [20]. As biologist Godfrey-Smith points out, biology, for instance, does not heavily rely on the concept of a law of nature, yet it remains a scientific discipline [8]. To advance the discussions on the scientific nature of cybersecurity, it is crucial to approach the field with rigorous scientific methodologies and approaches, starting with open datasets.

However, it is also important to note that cybersecurity today encompasses not only 'hard' sciences but also incorporates numerous components of the social sciences. It involves understanding the behavior of humans, particularly in the evolving digital ecosystem in which they now live. Social cybersecurity is now seen as a subdiscipline of cybersecurity and increasingly important for national security [2].

3 The Challenges of Cybersecurity Datasets

Today, as noted above, a significant component of cybersecurity research data relates to human behavior, and as such, it is subject to a variety of privacy considerations. In some cases, as human aspects become more prevalent, cybersecurity academic research is often considered “human subjects” research and becomes subject to review and approval by the institution’s Institutional

Review Board (IRB). As such, academic researchers, students and faculty are required to take training in behavioral and social research before beginning any contact with any human subjects engaged in their research. In addition, depending on the risks posed to the individuals involved in the research, limitations may be imposed on the way data is collected, stored, analyzed, and how and what results are made available to the public, usually in anonymized or aggregated formats. Many large research institutions such as the University of Virginia or the University of Tennessee do provide access to datasets that have been approved for public use by their institutions. However, such access is often subject to data use agreements in various scientific areas. However, many others do not, including ours.

Much of this cybersecurity research may also include the collection and analysis of personal identifiable information (PII) even if it is not directly obtained from individuals themselves or with their explicit consent. Examples of this include surveillance data. Handling such data appropriately is important in much of today's cybersecurity research. However, it becomes critical and often mandated in industries such as healthcare or finance, as well as governed by regulations such as the General Data Protection Regulation (GDPR) [1]. This is also further complicated by the differing privacy and data security regulations being implemented by various states including California, Virginia, and Massachusetts.

Another significant challenge for academic researchers is locating publicly available up-to-date data. Making such data widely accessible also presents the complication of potential exposure to hackers, including nation states, and its utilization by adversaries. Unlike other sciences, a significant portion of publicly available cybersecurity data cannot be directly applied to the current cybersecurity ecosystem. For example, some of the datasets available are from the early days of the Internet, reflecting an infrastructure largely based on perimeter defenses, such as firewalls and other defensive devices available at the time. Some open resources are sporadically updated but do not necessarily relate to the infrastructure of today and the current cybersecurity strategies such as Zero Trust [21].

Some doctoral students who were unable to find open datasets, turned to harvesting social media sources. The two most common cases have been Twitter and Reddit. These datasets have been primarily been used for textual analysis around disinformation campaigns [22].

Many publicly available cybersecurity related datasets require permission before they can be accessed. The notice ”Forbidden: you do not have access to this resource” or similar messages is discouraging for academic researcher who might not be given an option to request permission. Often, however, it means that the data source is no longer available. For example, the datasets previously made available by cybertrust.org are still listed on some datasets lists, even though the company was acquired by Verizon back in 2007 and its website no longer exists.

Another source of data is from commercial companies that publish routine summaries of cybersecurity threats and incidents. One major example is the Verizon Data Breach Investigations Report (DBIR) which uses data collected from over 80 organizations, including a mix of vendors and technology-related public and private sector entities. Verizon states that its data is available on GitHub, but a review of those "available" files does not enable a researcher to easily find specific datasets for further research and establish their generalizability for academic purposes. However, for many vendors providing cyber threat intelligence or data breach incident reports, their data is deemed proprietary.

Many of the open data sources include disciplines outside of cybersecurity. Consequently, the lack of a common cybersecurity nomenclature often makes finding the relevant datasets difficult and requires the use of numerous search terms. In certain instances, specific cybersecurity terminology, such as 'ransomware,' must be employed, while in others, synonyms for cybersecurity like 'computer security' or 'information assurance' are required.

Finally, datasets are often only known to a few, especially when collecting data from other disciplines. For instance, consider the case of a doctoral student working in the field of election security. Unable to find an open dataset for the specific data needed for analysis, the student resorted to surveying individuals listed as election officials. Close to an election, the student received a response from the first and only person who engaged, expressing the lack of time to complete the survey but offering to have a conversation instead. In this conversation, she directed him to a publicly available dataset, known primarily to the election official's community, which provided him an authoritative source for his research.

With many of our doctoral students coming from long careers in cybersecurity, their research interests have been on tracking the growing needs and changes of the cybersecurity workforce. Without a central professional organization monitoring the field, the students have created local datasets from job posting websites highlighting the knowledge, skills, and abilities needed in the field as well as the changes in the workforce over time.

4 Potential Solutions

4.1 Synthetic Data

Overcoming privacy and confidentiality issues is essential for many cybersecurity datasets which, by necessity, include PII or other sensitive information such as corporate financial information. In these cases, synthetic data may be considered a potential solution.

Generative Adversarial Networks (GANs), a type of neural network model, have gained in popularity in recent years due to the ability to generate realistic data. Their use for generating visual, numerical, and textual data has recently expanded rapidly. A GAN model is composed of a generator and a discriminator component. The generator creates data following examples from a training dataset. For each generated piece of data, the discriminator evaluates whether the generator's created data is real or fake. Together, this process is called adversarial training, where the generator produces increasingly realistic data to trick the discriminator while at the same time the discriminator is being trained on a real vs. generated dataset. The models are improving while trying to deceive or catch the opposing model [9]. This process increases the overall quality of the generated data to the point that it can trick the discriminator at a successful rate.

GANs are used in a wide range of applications from image and video generation to text-to-image creation, and in the newer fields of voice and music generation. In the past year, two examples have taken over popular culture. An image generation model, DALL-E and a large language model that can produce fluent language and solutions from user prompts, ChatGPT [18]. While these models have raised concerns regarding potential security threats, ranging from phishing to deep fakes by adversaries, they also have great potential to assist mainstream cybersecurity's primary concerns by enabling the production of realistic synthetic data and addressing data availability issues.

The Conditional Tabular GAN (CTGAN) is a variant of conditional GANs made primarily for generating synthetic data that is in line with the statistical properties of a training dataset [28]. The goal of these systems is often to protect the privacy of individuals in datasets through the generation of synthetic PII [14]. By generating synthetic data that is statistically similar to the original data, but does not contain any sensitive information, CTGAN can help security professionals protect the privacy of individuals while still allowing doctoral researchers access to modern, up-to-date datasets. The synthetic data generated by these models can enable institutions to release data for public analyses or provide research institutes with improved techniques to address the challenges of a fast-changing cybersecurity industry. [11] showed the efficacy of releasing datasets by replacing PII with synthetic data, while maintaining similar analytical results remaining similar.

Using state-of-the-art AI should not only be seen in the context of AI generated media but also these important security and privacy domains. Synthetic data creation can and should be a key component in any future cybersecurity ecosystem.

4.2 Central Curated Collection of Open Cybersecurity Datasets

A small number of cybersecurity datasets are available from “collections” sites. One example is the IEEE DataPort, however, cybersecurity is not recognized as a dataset category, such as artificial intelligence or cloud computing. In addition, a monthly individual or institution subscription is required. Other dataset repositories such as Kaggle, GitHub, and the Registry of Open Data on AWS are very broad in terms of their content, making it challenging for academics to identify relevant cybersecurity datasets suitable for their academic research.

We need to look beyond these traditional computer science related resources. One example to consider as a model is the Inter-university Consortium of Politics and Social Science Research (ICPSR), run by the University of Michigan. It goes back to 1962 when 25 universities were engaged. Today there are more than 750 academic institutions and research organizations collaborating in the consortium. Currently more than 70,000 datasets have been curated and are available on demand. There are also a smaller number of datasets which include at least one restricted file. The collection is well-organized and encompasses 21 specialized areas such as criminal justice and terrorism, not so distant from cybersecurity.

5 Next Steps

The authors believe that the availability of curated datasets is important both for our doctoral program and for our own faculty research. First, we need to ensure that making the data generated by our faculty and students publicly available does not violate our institution’s IRB requirements. We will collaborate with them to develop policies and procedures similar to those followed by other academic institutions, whether actual data or synthetic data.

We have recently joined the ICPSR consortium and plan to use this resource to publish our datasets, and hopefully we can work with them to create a cybersecurity specialty. At the same time, we plan to experiment with synthetic dataset generation, in particular using CTGAN methods to take datasets with PII data and generating new datasets which can be used in replication studies to demonstrate their usefulness.

Our next project is to develop a Working List of Existing Datasets that can be utilizedd in our academic research initiatives. We have initiated a small project using web scraping techniques to access existing repositories and identify candidate datasets. These dataset will be tagging with a consistent cybersecurity nomenclature, which is currently under development. At the same time, we will develop a notification process where faculty and students can contribute to our collection of cybersecurity datasets by informing us of the datasets they find in their research for later evaluation. We will also develop automated validation tools to continually evaluate the availability of the datasets on our internal list. This will be run monthly and faculty and students will evaluate the issue and remediate by either deleting the issue or by updating the link and other information.

Finally, we will evaluate the impact of a usable list of cybersecurity datasets on our doctoral student's research output, including replication initiatives.

References

[1] Regulation (eu) 2016/679 of the european parliament and of the council. *Official Journal of the European Union*, L119:1–88, 2016.

[2] David M. Beskow and Kathleen M. Carley. Social cybersecurity: An emerging national security requirement. *Military review*, 99:117, 2019.

[3] Andy Cockburn, Pierre Dragicevic, Lonni Besançon, and Carl Gutwin. Threats of a replication crisis in empirical computer science. *Commun. ACM*, 63(8):70–79, jul 2020.

[4] W. Crumpler and A. Lewis, J. The cybersecurity workforce gap. `https://csis-website-prod.s3.amazonaws.com/s3fs-public/publication/190129_Crumpler_Cybersecurity_FINAL.pdf`, 2019.

[5] Josiah Dykstra. *Essential Cybersecurity Science.* O'Reilly Media, Inc., 2015.

[6] Thomas W. Edgar. *Research methods for cyber security / Thomas W. Edgar, David O. Manz.* Syngress, an imprint of Elsevier, Cambridge, MA, 2017.

[7] Gloria González Fuster and Lina Jasmontaite. *Cybersecurity Regulation in the European Union: The Digital, the Critical and Fundamental Rights*, pages 97–115. Springer International Publishing, Cham, 2020.

[8] Peter Godfrey-Smith. *Theory and Reality: An Introduction to the Philosophy of Science.* University of Chicago Press, 1st edition, 2003.

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Y. Bengio. Generative adversarial networks. *Advances in Neural Information Processing Systems*, 3, 06 2014.

[10] Cormac Herley and P.C. Van Oorschot. Sok: Science, security and the elusive goal of security as a scientific pursuit. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 99–120, 2017.

[11] Anantaa Kotal, Aritran Piplai, Sai Sree Laya Chukkapalli, and Anupam Joshi. Privetab: Secure and privacy-preserving sharing of tabular data. In *Proceedings of the 2022 ACM on International Workshop on Security and Privacy Analytics*, IWSPA '22, page 35–45, New York, NY, USA, 2022. Association for Computing Machinery.

[12] Carl E. Landwehr. Cybersecurity: From engineering to science. *The Next Wave: The National Security Agency's review of emerging technologies*, 19(2), 2012.

[13] Alessandro Marotta and Stuart Madnick. Analyzing the interplay between regulatory compliance and cybersecurity. Working Paper CISL# 2020-06, 2020.

[14] Robert Mayer, Michael Hittmeir, and Andreas Ekelhart. Privacy-preserving anomaly detection using synthetic data. In *Data and Applications Security and Privacy XXXIV: 34th Annual IFIP WG 11.3 Conference, DBSec 2020, Regensburg, Germany, June 25–26, 2020, Proceedings*, volume 34, pages 195–207. Springer International Publishing, 2020.

[15] Héctor D. Menéndez, Sukriti Bhattacharya, David Clark, and Earl T. Barr. The arms race: Adversarial search defeats entropy used to detect malware. *Expert Syst. Appl.*, 118:246–260, 2019.

[16] Tyler Moore, Erin Kenneally, Megan Collett, and Prakrati Thapa. Valuing cybersecurity research datasets. In *18th Workshop on the Economics of Information Security (WEIS)*, 2019.

[17] National Academy of Sciences. *Reproducibility and Replicability in Science.* National Academy of Sciences Press, 2019.

[18] OpenAI. Chatgpt. Retrieved Month 6/2/2023, 2023.

[19] Sean Peisert, Bruce Schneier, Hamed Okhravi, Fabio Massacci, Terry Benzel, Carl Landwehr, Mohammad Mannan, Jelena Mirkovic, Atul Prakash, and James Bret Michael. Perspectives on the solarwinds incident. *IEEE Security Privacy*, 19(2):7–13, 2021.

[20] R. Ramachandran. The science of cybersecurity and its future challenges. `https://www.isaca.org/resources/news-and-trends/isaca-now-blog/2023/the-science-of-cybersecurity-and-its-future-challenges`, 2023.

[21] Stephen Rose, Oliver Horchert, Steven Mitchell, and Sean Connelly. Zero trust architecture. Technical Report Special Publication 800-207, National Institute of Standards and Technology (NIST), 2020.

[22] Vinay Setty and Erlend Rekve. Truth be told: Fake news detection using user reactions on reddit. In *Proceedings of the 29th ACM International Conference on Information amp; Knowledge Management*, CIKM '20, page 3325–3328, New York, NY, USA, 2020. Association for Computing Machinery.

[23] Mark F. Tardiff, George T. Bonheyo, Katherine A. Cort, Thomas W. Edgar, Nancy J. Hess, William J. Hutton, Erin A. Miller, Kathleen E. Nowak, Christopher S. Oehmen, Emilie A. H. Purvine, Gregory K. Schenter, and Paul D. Whitney. Applying the scientific method to cybersecurity research. In *2016 IEEE Symposium on Technologies for Homeland Security (HST)*, pages 1–8, 2016.

[24] The MITRE Corporation. Science of cyber-security. `https://irp.fas.org/agency/dod/jason/cyber.pdf`, 2010.

[25] The Royal Society. Progress and research in cybersecurity. `https://royalsociety.org/topics-policy/projects/cybersecurity-research/`, 2016.

[26] The White House. National cybersecurity strategy. `https://www.whitehouse.gov/wp-content/uploads/2023/03/National-Cybersecurity-Strategy-2023.pdf`, 2023.

[27] Ivan Trenchev, Willian Dimitrov, Georgi Dimitrov, Tanya Ostrovska, and Miglena Trencheva. Mathematical approaches transform cybersecurity from protoscience to science. *Applied Sciences*, 13(11), 2023.

[28] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional gan. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[29] Muwei Zheng, Hannah Robbins, Zimo Chai, Prakash Thapa, and Tyler Moore. Cybersecurity research datasets: Taxonomy and empirical analysis. In *11th USENIX Workshop on Cyber Security Experimentation and Test (CSET 18)*, Baltimore, MD, August 2018. USENIX Association.

Analyzing the Impact of Summer Orientation Programs for Women Computing Undergraduates*

Mary Villani[1], Ilknur Aydin[1], Lisa Cullington[2]
[1]Computer Systems
Farmingdale State College, SUNY
Farmingdale, NY 11735
{villanmv, aydini}@farmindale.edu
[2]Learning Experience
National University, CA 92123
lcullington@nu.edu

Abstract

Consistent with nationwide enrollment, Farmingdale State College experiences a significant gender disparity in its undergraduate computing degrees. Previous research has demonstrated that a strong sense of belonging and academic self-concept can have positive impacts on student retention and persistence in STEM higher education, particularly for underrepresented students. This study examines the impact of participation in a summer orientation program developed to build camaraderie among women students and improve their sense of belonging and academic self-concept. Preliminary results from pre- and post-orientation surveys (measuring immediate impact) and end of semester surveys (measuring longer term impact) show generally positive results.

1 Introduction

Farmingdale State College is primarily a commuter college with approximately 10,000 enrolled, where only 10% live on campus. The overall student body is 42% women, 47% minority, 62% financial aid recipients, and 82% full- or part-time employees. Students in computing majors share the overall college demographic; except that the percentage of women students has hovered between 8-15% over a decade (Fig. 1). The Computer Systems Department has offered Computer Programming & Information Systems (CPIS) degree for over twenty years and launched the Computer Science (CS) degree in fall 2021. Though attempts to address the gender disparity in these degree programs were made from 2011 to 2019, these efforts did not last [15]. Beginning spring 2020, a new set of initiatives (support programs) for women computing students started by the authors which included maintaining an active ACM-W Supporting Women in Computing Club (SWiC) by working with student leaders [16], taking students to women-centric conferences [17], and organizing and hosting summer orientation programs for women computing students [17]. For the purposes of this research, "women" signifies participants who identified as a woman, inclusive of cisgender and transgender.

Despite an increased enrollment of women in overall undergraduate STEM programs over the last ten years, gender disparity in enrollment in the mathematical and computer sciences has continued to persist [12]. As of 2020, men have earned over 70% of all bachelor's degrees in mathematical and computer sciences and engineering [12]. Given this gender disparity, educational researchers have studied the impacts of initiatives aimed at supporting women in STEM. Often these initiatives investigate student motivation for majoring in STEM, students' sense of belonging and connection to the institution and/or the STEM field [13, 11] and students' perceptions of their own academic abilities within the STEM field or their academic self-concept [3]. Building off this research, this paper focuses on the summer orientation initiative, which was held for incoming and returning women computing students, to help build camaraderie and empower women prior to start of each academic year. The research goals are to measure and examine the impact of the orientation program on women students' 1) social and academic experiences as they complete their degree programs, 2) sense of belonging, and 3) academic self-concept.

2 Literature and Theoretical Framework

Involvement in the academic and social opportunities provided by an undergraduate institution has been long established as effective ways for students to feel a part of the college community [1]. Student persistence is positively

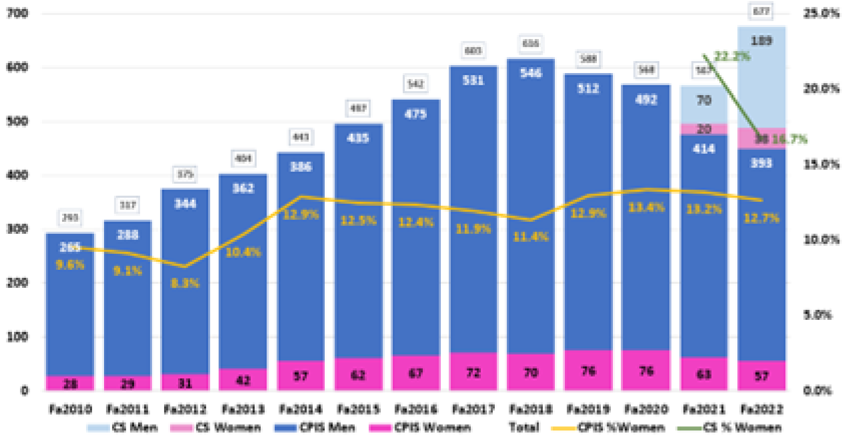


Figure 1: Enrollment by gender and degree program at Farmingdale State College

impacted by enhanced student involvement in curricular and co-curricular opportunities. Based on a review of the relevant literature, the following theoretical framework was developed to analyze the impact of orientation program for women in undergraduate computing programs.

Sense of belonging has been studied by educational researchers in higher education settings. Positive social engagement cultivates a strong sense of belonging [7]. The extent to which students feel connected to their academic institutions is associated with many positive academic outcomes such as persistence and retention [14, 13]. Researchers have highlighted the importance of engaging with peers and faculty [14]. Friendships with peers [5, 6]; and close ties with faculty mentors [10] has been demonstrated to positively impact a student's persistence in the major and college, overall.

Students engage more deeply in the academic experiences of an institution as their academic self-concept increases. Academic self-concept identifies students' perception of their academic abilities and understanding of themselves as having an academic identity. Students with elevated levels of academic self-concept tend to have more positive academic outcomes in areas such as persistence, retention, and academic achievement [9]. In their seminal work in researching self-concept, [4] demonstrate that interaction with one's environment especially impacts one's self-concept. Students' engagement with peers and faculty provides an opportunity for social comparison and a frame of reference in which to judge one's academic abilities. Recent work has demonstrated the importance of a strong academic self-concept for sustaining college retention.

Based on the review of relevant literature, the following goal was established for this study: measure the effectiveness and impact the summer orientation program on women computing students social and academic experience, sense of belonging, and academic self-concept. In particular, the following research questions were identified:

Question 1) What is the immediate/short-term impact of attending the orientation program on women participants?

Question 2) What is longer term impact on social experience and sense of belonging of women participants?

Question 3) What is longer term impact on academic experience and academic self- concept of women participants?

3 Positionality Statement

The authors provide a diverse set of experiences that inform the research process related to this study. While one of the authors identifies as a woman with a business/industry computer consulting background prior to her academic career, the other identifies as a woman in the computing and engineering field who shares a similar ethnic background as some of the student participants. Both faculties have firsthand experience of the challenges found in navigating male-dominated fields in both academia and industry. The third author identifies as a woman educational researcher with personal experience as a first-generation college student, again sharing similar background with some of the student participants. While this study is not specifically focused on first-generation college students, the author brings a positionality to the work that highlights the importance of advocating for underrepresented students in academic spaces. As a collective, the authors' positionality puts forward a belief in creating supportive communities for women computing students at Farmingdale State College. By implementing initiatives to create these communities, women computing students will experience an increased sense of belonging, improve academic self-concept, and contribute to a more diverse community at Farmingdale State College and the computing field, more broadly.

4 Research Design and Methodology

Authors have conducted two summer orientation programs so far. The first cohort was in August 2021 (n=21 attended) returning to campus following COVID19 pandemic [2] and the second cohort was in August 2022 (n=32 attended). All women computing students in the target group were invited to both orientations (so some of the participants could return from previous year). The target group included only the CPIS and CS students in 2021 summer orientation and then was extended to include four computing majors with the addition of Computer Security Technology (CST) and Security Systems Technology (SST) majors in Summer 2022. Pre- and post- orientation surveys were collected per orientation to measure its immediate impact on attendees. Longer-term impact was measured with the end of semester surveys collected in December 2021, May 2022, and December 2022, so far. The results of 2022 pre- and post-orientation surveys, their comparison to the inaugural summer 2021 pre- and post-surveys, and three end of semester survey are analyzed, and findings are shared in Section 5, below.

4.1 Design and Delivery of Summer 2022 Orientation

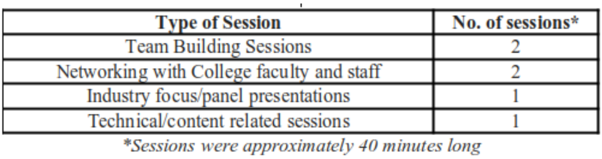
The objectives of the (re)orientation program was to (re)acquaint women students with their peers and faculty, provide bonding opportunities using team building activities and games, deliver technical sessions to empower the women students for the upcoming academic year, as well as to provide inspiration to the women students through the connections made with other women in the program. Therefore, the design of the second summer orientation in 2022 began with creating an agenda with these objectives in mind and accordingly from the lessons learned from executing previous 2021 program [2]. Both orientations were designed as a day program starting in the morning and ending in the late afternoon.

SWiC student leaders were involved and collaborated with the authors both in designing the day activities and delivering them (such as welcoming participants, overseeing the team building activities). Table topics during breakfast and lunch were used as icebreaker activities where upper-class and new incoming students were deliberately mixed up to sit at the tables and organized into the teams during the games. Four other men faculties (two from CPIS/CS and other two from CST/SST security department) also were involved in the day by delivering the two technical workshops, and the panel discussion about the majors. A summary of the agenda is shown in Table 1.

4.2 Data Collection

Pre- and post-orientation survey data collection was part of the welcome package where the QR codes were provided enabling students to take the surveys from their cellphones while waiting for the program to start and at the end of the day, respectively. Note that respondents were allowed to select more than one option about participant expectations in many questions to provide

Table 1: Summary of Orientation Day Agenda



Type of Session	No. of sessions*
Team Building Sessions	2
Networking with College faculty and staff	2
Industry focus/panel presentations	1
Technical/content related sessions	1

**Sessions were approximately 40 minutes long*

an explanation where number of responses exceed the number of participants in the survey.

The end of semester survey was distributed via email at the end of the fall 2021, spring 2022, and fall 2022 semesters. Emails were sent requesting participation by two SWiC faculty advisors (authors), spread out over a two-week period to elicit maximum participation. Qualtrics was used to create, administer, and collect survey information from the target group.

5 Results and Findings

5.1 Target Group Demographic

The summer orientation programs were open to both returning and new incoming women students due to low numbers in each category and concerns that commuting and working invitees would not be able to attend.

The inaugural 2021 summer orientation was timed with the return to campus after two and a half semesters of being remote after the global COVID19 pandemic. Of the 78 women invited (59 CPIS and 19 CS majors), 32 students RSVPed, 22 attended and 19/22 took the pre/post orientation surveys. Attendance was all women students, half newly incoming and half returning CPIS/CS students. In 2022 orientation, CST and SST majors were also included (students in all four majors take some of their required courses together). Of the 122 women students invited (51 CPIS, 36 CS, 17 SST, 18 CST majors), 53 students RSVPed, 32 showed up and, 29 and 28 students took the pre/-post orientation surveys, respectively. Of the 29 attendees, 25 were women students, 3 men-identifying students (they were invited as leaders of the co-ed ACM Computer tech student club to represent the club), and 1 transgender student. The attendees in 2022 were almost equal percentages from each level of study (27% first year, 24sophomore, 31% junior and 17% senior). CPIS and CS students formed most of the participants, about 43% and 37%, respectively, compared to the CST/SST students, about 10% from each. Figure 2 shows the diversity of the participants in race and ethnicity (14% white) which is consistent with the Farmingdale State College demographic.

5.2 Pre-Survey: Expectations of Participants

Pre-Survey questions focused on the expectation and motivation of the participants in attending the orientation programs. The data in Table 2 suggest that making connections with other women students and faculty is the most popular motivation.

5.3 Post-Survey: Immediate Impact (Question 1)

Students were surveyed in pre- and post-surveys about their awareness of gender imbalance in computing. Table 3 shows how participants' perspective demonstrated minimal changes from pre- to post-survey, except that surprisingly in 2022 orientation participants perceived the field less male-dominated post-orientation. A possible explanation for these responses might be an impact from increased connection with other women computing majors during the all-day orientation.

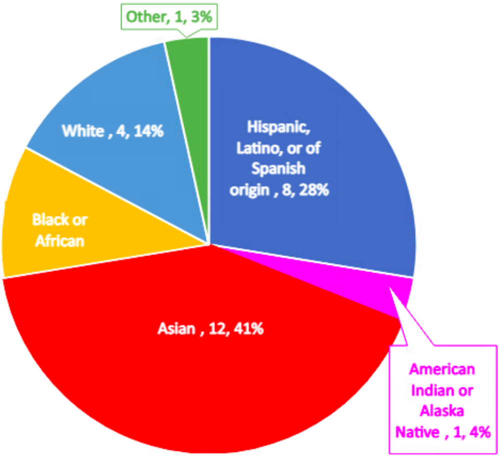
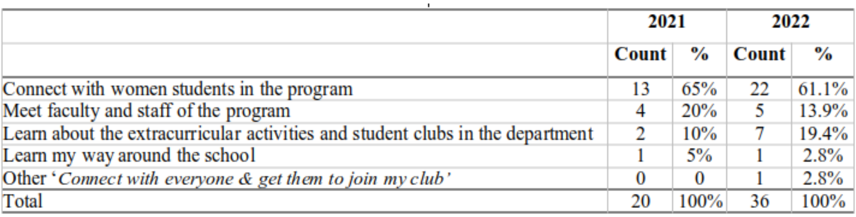


Figure 2: Race/Ethnicity of 2022 Orientation Attendees

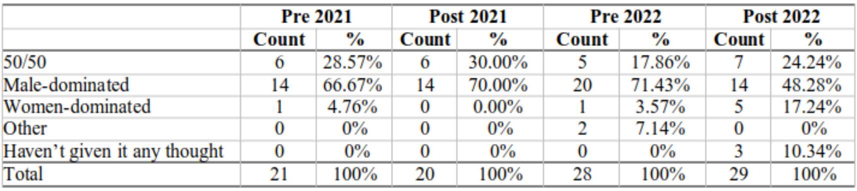
When attendees were asked how it made them feel to learn about the gender disparity post-orientation, the majority were "inspired to succeed in computing classes to defy the gender gap" (Table 4). Additional information obtained from post-surveys showed other positive impacts including more than 85% said "it was easy to connect with others" in both years, 60% of 2021 and 45% of 2022 participants said, ”they would get involved in SWiC club”, and 90% of 2021 and 82% of 2022 participants responded, “it was time well spent” (which shows satisfaction of the participants with the orientation day).

Table 2: Pre-Orientation responses about participants’ expectations



	2021		2022	
	Count	%	Count	%
Connect with women students in the program	13	65%	22	61.1%
Meet faculty and staff of the program	4	20%	5	13.9%
Learn about the extracurricular activities and student clubs in the department	2	10%	7	19.4%
Learn my way around the school	1	5%	1	2.8%
Other ‘*Connect with everyone & get them to join my club’*	0	0	1	2.8%
Total	20	100%	36	100%

Table 3: Pre- and Post-orientation Perception of Gender Imbalance

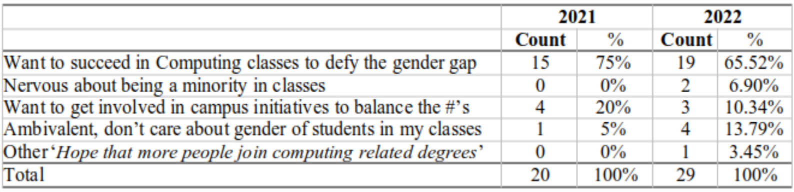


	Pre 2021		Post 2021		Pre 2022		Post 2022	
	Count	%	Count	%	Count	%	Count	%
50/50	6	28.57%	6	30.00%	5	17.86%	7	24.24%
Male-dominated	14	66.67%	14	70.00%	20	71.43%	14	48.28%
Women-dominated	1	4.76%	0	0.00%	1	3.57%	5	17.24%
Other	0	0%	0	0%	2	7.14%	0	0%
Haven’t given it any thought	0	0%	0	0%	0	0%	3	10.34%
Total	21	100%	20	100%	28	100%	29	100%

Tables 5 and 6 show positive feeling and attitude by women students such as motivation to attend classes and confidence to succeed post-orientation. When prompted to rate their ability to connect with other women students in the program because of attending the orientation program, the response was positive and even improved in 2022. While student responses of confidence to succeed increased in 2022, the overall levels of confidence remained low. This could be explained by the content of the orientation. The orientation focused more on building a community and networking than on technical skills. Future research is needed to determine if a change in the content might yield higher results in students’ level of confidence to succeed.

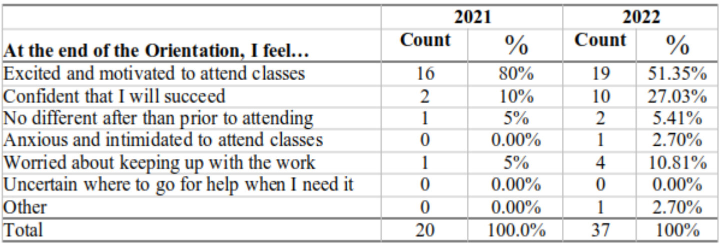
Overall, post-orientation survey data from two cohorts in 2021 and 2022 (Tables 3-4) demonstrate the positive short-term impact on women participants from attending the women-focused orientation program. Specific elements of the orientation program (examples team building exercises, scavenger hunt, technical hands-on exercises, and table topic discussions) provided its participants opportunities for social and academic experiences, forming friendships

Table 4: Post-survey responses on impact about gender imbalance



	2021		2022	
	Count	%	Count	%
Want to succeed in Computing classes to defy the gender gap	15	75%	19	65.52%
Nervous about being a minority in classes	0	0%	2	6.90%
Want to get involved in campus initiatives to balance the #'s	4	20%	3	10.34%
Ambivalent, don't care about gender of students in my classes	1	5%	4	13.79%
Other '*Hope that more people join computing related degrees*'	0	0%	1	3.45%
Total	20	100%	29	100%

Table 5: Measure of outlook post-orientation program



	2021		2022	
At the end of the Orientation, I feel...	Count	%	Count	%
Excited and motivated to attend classes	16	80%	19	51.35%
Confident that I will succeed	2	10%	10	27.03%
No different after than prior to attending	1	5%	2	5.41%
Anxious and intimidated to attend classes	0	0.00%	1	2.70%
Worried about keeping up with the work	1	5%	4	10.81%
Uncertain where to go for help when I need it	0	0.00%	0	0.00%
Other	0	0.00%	1	2.70%
Total	20	100.0%	37	100%

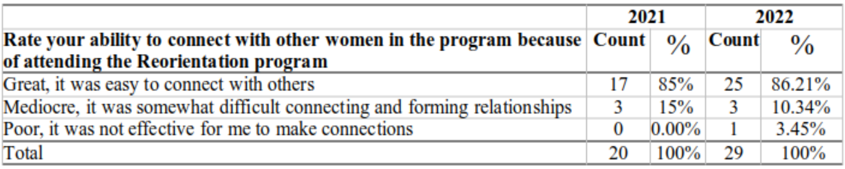
with peers, and receiving faculty support and mentorship. These elements have been correlated with positive academic outcomes for student participants by previous research [10, 14, 7, 5].

5.4 End of Semester: Longer Impact (Questions 2 & 3)

Table 7 summarizes response rate for the three end of semester surveys and number of respondents that attended the orientations. In end-of-semester surveys, participants who responded that they attended a summer orientation program were asked to select which best described the impact. Table 8 shows that students felt less intimidated in classes and more confident in attending classes (because of seeing familiar student and faculty faces from the orientation programs). In addition to the data in Table 8, 62% end of fall'21, 57% end of spring'22, and 65% end of fall'22 survey participants responded that they became involved in SWiC on campus because of attending orientation.

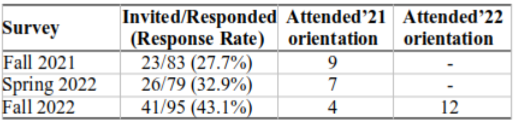
Tables 9 and 10 show women students' perception of academic experience (can be linked to academic self-concept) and social experience (can be linked to sense of belonging) while completing their degree programs, respectively, as measured in the three end of semester surveys and for those participants

Table 6: Impact of attending reorientation on making connections



	2021		2022	
Rate your ability to connect with other women in the program because of attending the Reorientation program	**Count**	**%**	**Count**	**%**
Great, it was easy to connect with others	17	85%	25	86.21%
Mediocre, it was somewhat difficult connecting and forming relationships	3	15%	3	10.34%
Poor, it was not effective for me to make connections	0	0.00%	1	3.45%
Total	20	100%	29	100%

Table 7: End of Semester Survey Response Rate



Survey	**Invited/Responded (Response Rate)**	**Attended'21 orientation**	**Attended'22 orientation**
Fall 2021	23/83 (27.7%)	9	-
Spring 2022	26/79 (32.9%)	7	-
Fall 2022	41/95 (43.1%)	4	12

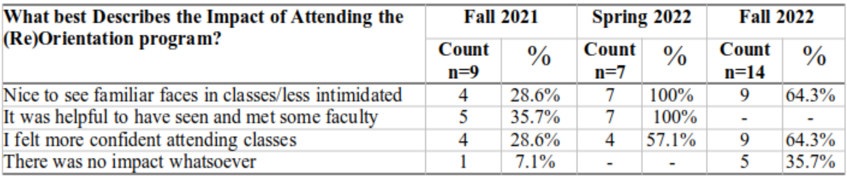
that attended an orientation program vs. 'all' (includes students that did not attend an orientation program). The academic experience of women students who attended the orientation program does not exceed responses from 'all' women students except slightly in end of spring'22 when combining excellent and good ratings.

Similarly, social experience of women students who attended an orientation program compares slightly worse compared to 'all' except in the last fall'22 end of semester survey. Additional data collection may shed light on these responses and help determine the impacts of the orientation program on the social experience of women students.

6 Discussion and Limitations

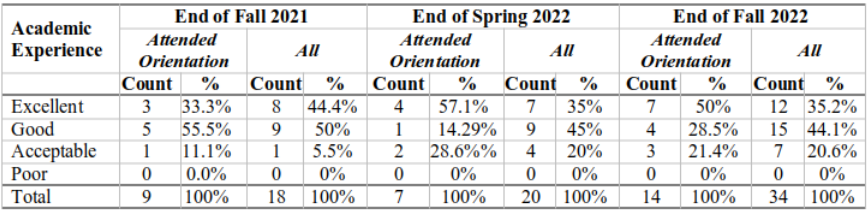
The results and findings presented above in general show positive outcomes of the women in computing initiatives underway since 2019 and specifically the two orientation programs that was held in summer 2020 and 2021. However, the authors acknowledge the following limitations regarding the survey participant pool: (i) the results and findings reported are from a small number of women survey participants as the women numbers are low in Farmingdale State College computing majors!, (ii) the participants of the orientation represent a small percentage of the invitees (28%, n=22/78 in Summer'21 and 26%, n=32/122 in Summer'22) as the attendance is voluntary, and (iii) although the second orientation cohort (Summer'22), included CST/SST majors, the three end of

Table 8: End of Semester Responses of orientation attendees



What best Describes the Impact of Attending the (Re)Orientation program?	Fall 2021		Spring 2022		Fall 2022	
	Count n=9	%	Count n=7	%	Count n=14	%
Nice to see familiar faces in classes/less intimidated	4	28.6%	7	100%	9	64.3%
It was helpful to have seen and met some faculty	5	35.7%	7	100%	-	-
I felt more confident attending classes	4	28.6%	4	57.1%	9	64.3%
There was no impact whatsoever	1	7.1%	-	-	5	35.7%

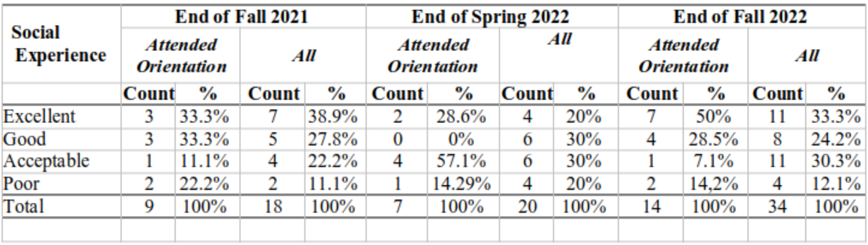
Table 9: End of Semester Responses about Academic Experience



Academic Experience	End of Fall 2021				End of Spring 2022				End of Fall 2022			
	Attended Orientation		*All*		*Attended Orientation*		*All*		*Attended Orientation*		*All*	
	Count	%	Count	%	Count	%	Count	%	Count	%	Count	%
Excellent	3	33.3%	8	44.4%	4	57.1%	7	35%	7	50%	12	35.2%
Good	5	55.5%	9	50%	1	14.29%	9	45%	4	28.5%	15	44.1%
Acceptable	1	11.1%	1	5.5%	2	28.6%%	4	20%	3	21.4%	7	20.6%
Poor	0	0.0%	0	0%	0	0%	0	0%	0	0%	0	0%
Total	9	100%	18	100%	7	100%	20	100%	14	100%	34	100%

the semester surveys conducted so far were only administered on CPIS/CS majors. Combined CST/SST majors has been smaller a group (31% of all women computing students in fall'22) compared to the combined CPIS/CSC majors, but the plan is to include all the four computing majors in future end of semester surveys. The study presented a limitation in terms of data collection methods: the quantitative survey's connection to and operationalization of the conceptual framework and application to such a small sample size. Initially guided by administrative data collection purposes, this study can shed light on critical insights for future research. As such, the authors have considered incorporating established survey measures more intricately connected to sense of belonging and academic self-concept, as well as focus groups, and/or interviews to elicit further understandings of the nature of such an experience for the small group of women computing students. Lastly, this work is in progress and the findings are from a short period of time (three semesters). The work is approved by college IRB for 5 years; hence, future end of semester surveys will be conducted for further data collection.

Table 10: End of Semester Responses about Social Experience



Social Experience	**End of Fall 2021**				**End of Spring 2022**				**End of Fall 2022**			
	Attended Orientation		***All***		***Attended Orientation***		***All***		***Attended Orientation***		***All***	
	Count	**%**	**Count**	**%**	**Count**	**%**	**Count**	**%**	**Count**	**%**	**Count**	**%**
Excellent	3	33.3%	7	38.9%	2	28.6%	4	20%	7	50%	11	33.3%
Good	3	33.3%	5	27.8%	0	0%	6	30%	4	28.5%	8	24.2%
Acceptable	1	11.1%	4	22.2%	4	57.1%	6	30%	1	7.1%	11	30.3%
Poor	2	22.2%	2	11.1%	1	14.29%	4	20%	2	14,2%	4	12.1%
Total	9	100%	18	100%	7	100%	20	100%	14	100%	34	100%

7 Conclusions and Future Work

This study examines the impact of participation in a summer orientation program developed to build camaraderie among women students and improve their sense of belonging and academic self-concept. Preliminary immediate results from pre- and post-orientation surveys and end of semester surveys from three semesters indicate positive outcomes. Future surveys from upcoming semesters are planned to include qualitative study to enhance the analysis of the initiatives' impact.

References

[1] Alexander W Astin. "Student involvement: A developmental theory for higher education." In: (1999).

[2] Ilknur Aydin, Mary Villani, and Lisa Cullington. "Designing a (Re) Orientation program for Women Computing Students at a Commuter College and Measuring Its Effectiveness". In: *2022 ASEE Annual Conference & Exposition*. 2022.

[3] Amy R Betz et al. "Improving academic self-concept and stem identity through a research immersion: Pathways to STEM summer program". In: *Frontiers in Education*. Vol. 6. Frontiers Media SA. 2021, p. 674817.

[4] Mimi Bong and Einar M Skaalvik. "Academic self-concept and self-efficacy: How different are they really?" In: *Educational psychology review* 15 (2003), pp. 1–40.

[5] Jessica A Crowe. "Creating a departmental climate that increases a student's sense of belonging, perceived faculty support, and satisfaction with the major". In: *Innovative Higher Education* 46.1 (2021), pp. 95–109.

[6] Glenn M Davis et al. "Students' sense of belonging: The development of a predictive retention model". In: *Journal of the Scholarship of Teaching and Learning* 19.1 (2019).

[7] Cari Gillen-O'Neel. "Sense of belonging and student engagement: A daily study of first-and continuing-generation college students". In: *Research in Higher Education* 62.1 (2021), pp. 45–71.

[8] Abdulkadir Haktanir et al. "Resilience, academic self-concept, and college adjustment among first-year students". In: *Journal of College Student Retention: Research, Theory & Practice* 23.1 (2021), pp. 161–178.

[9] Leonie Jacob, Andreas Lachner, and Katharina Scheiter. "Do school students' academic self-concept and prior knowledge constrain the effectiveness of generating technology-mediated explanations?" In: *Computers & Education* 182 (2022), p. 104469.

[10] Ernest T. Pascarella. "Student-Faculty Informal Contact and College Outcomes". In: *Review of Educational Research* 50.4 (1980), pp. 545–595. DOI: 10.3102/00346543050004545. eprint: https://doi.org/10.3102/00346543050004545. URL: https://doi.org/10.3102/00346543050004545.

[11] Katherine Rainey et al. "Race and gender differences in how sense of belonging influences decisions to major in STEM". In: *International journal of STEM education* 5 (2018), pp. 1–14.

[12] National Center for Science and Engineering Statistics (NCSES). "Diversity and STEM: Women, Minorities, and Persons with Disabilities 2023". In: (2023).

[13] Terrell L Strayhorn. *College students' sense of belonging: A key to educational success for all students*. Routledge, 2018.

[14] Vincent Tinto. *Completing college: Rethinking institutional action*. University of Chicago Press, 2012.

[15] Mary Villani and Ilknur Aydin. "Learning from the Journey: A Decade of Supporting Women in Computing at a Commuter State College". In: *2021 Conference on Research in Equitable and Sustained Participation in Engineering, Computing, and Technology (RESPECT)*. IEEE. 2021, pp. 1–2.

[16] Mary Villani and Ilknur Aydin. "Mentoring a Women in Computing Student Club: The Good, The Bad, and The Ugly". In: *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education V. 2*. 2022, pp. 1191–1191.

[17] Mary V Villani, Ilknur Aydin, and Lisa Cullington. "Analyzing the Impact of Attending a Women in Computing Conference on Undergraduate Computing Students". In: *2023 ASEE Annual Conference & Exposition*. 2023.

Exploring ChatGPT's Ability to Solve Programming Problems with Complex Context[*]

Nghia D. Tran[1], James J. May[2], Nguyen Ho[3], Linh B. Ngo[2]
[1]Wesley College, Melbourne, VIC 3182, Australia
nghia.tran@wesleycollege.edu.au
[2]West Chester University, West Chester PA
{jm1006779,lngo}@wcupa.edu
[3]Loyola University Maryland, Baltimore MD
tnho@loyola.edu

Abstract

This paper presents a preliminary study on ChatGPT's ability to generate a working solution from a complex programming problem's textual description. Utilizing an online competitive programming platform's problem statements and its respective difficulty measures, we were able to examine ChatGPT's capabilities using the platform's solution status as a performance indicator. The experimental results show a strong relationship between the problem's perceived difficulty level, as provided by the platform, and the final solution status. Various techniques were used to measure the readability level of the problems' text, and we also found statistical relationship among several of them regarding the final status. The results also hint at a potential limitation of ChatGPT to understand complex programming problem context.

1 Introduction

The popularization of AI-enabled tools to support code development such as GitHub's Copilot and ChatGPT raises the possibility of AI having the ability to do the job of a software developer. GitHub's Copilot provides suggestions in the form of code snippets as users write their own code. ChatGPT, based on what users inquire, is able to generate code and explain what the algorithm is doing with a lot of detail. Researchers have begun to investigate the long-term impacts and implications of these AI-enabled tools on the labor market, including software development jobs. One working paper, for example, finds that information processing industries will be exposed to "high economic impact without distinguishing between labor-augmenting or labor-displacing effects" [6].

In this work, we aim to empirically explore the ability of one of these tools, ChatGPT in this case, to understand and solve complex programming problems. More specifically, we explore ChatGPT's capability to infer the underlying context hidden within the complex textual content of a problem rather than to simply provide a solution to a direct algorithmic question. Furthermore, we investigate whether the textual contents and readability of problem statements contribute to the accuracy of ChatGPT's generated solutions.

The rest of the paper is as follows. Section 2 describes the background information needed to understand what ChatGPT is, how we quantify and analyze the text of a programming problem, and what a competitive programming platform is and what platform we used for our study. In section 3, we describe the experimental setup and the data collection process. The results are examined in section 4. Section 5 concludes the paper and discusses future work.

2 Background

2.1 ChatGPT

ChatGPT is a large-scale conversational language model based on the GPT-3.5 architecture developed by OpenAI and made available publicly in November 2022 [2]. As a large language model, it generates human-like responses to a wide range of user prompts, including question answering, dialogue generation, and text completion. ChatGPT is pre-trained, mainly through supervised and reinforcement learning mechanisms, on a massive corpus of text data and fine-tuned on specific tasks and technologies [15]. This allows ChatGPT to produce natural-sounding language and contextually relevant responses. The dynamic architecture of ChatGPT is based on a transformer neural network capable of understanding patterns and relationships between words and phrases [9]. This enables the language model to process a diversity of complex inputs and outputs as well as generate coherent and grammatically correct text. Additionally, it leverages techniques such as attention mechanisms and positional encoding to improve its comprehension and generation of language [5]. Having the potential to transform human interactions with artificial intelligence and automate numerous aspects of our daily lives, ChatGPT is a powerful, revolutionary tool for research, education, and entertainment.

2.2 Text Analysis

A useful source of English readability indices is the Python *textstat* library [1]. We studied all 11 different systems and formulas of text analysis implemented by *textstat* to quantify the readability of the problem statements presented to ChatGPT: the Automated Readability Index, Dale-Chall Readability, Flesch Reading Ease, Flesch-Kincaid Grade, SMOG Index, Coleman-Liau Index, Linsear Write Formula, Gunning Fog, Lexicon Count, Text Standard, and Difficult Words.

The *Automated Readability Index (ARI)* is a formula designed for assessing the level of readability for a written text to help professionals evaluate the complexity of English texts. It considers both the average number of words per sentence and the average number of characters per word in a given text [16]. These two factors are integrated to produce a score that represents the grade level of the text.

The *Dale-Chall Readability (DCR)* is a common readability formula for measuring the grade-level difficulty of reading materials. The DCR is computed based on a formula incorporating textual elements such as sentence length, use of proper nouns, and word familiarity. The formula takes into account the average sentence length in words and the percentage of unfamiliar and difficult words that are not included in the devised Dale-Chall list of 3000 common words considered familiar to fourth-graders [4]. The resulting score is a grade level that reflects the minimum education level needed to understand the text.

The *Flesch Reading Ease (FRE)* formula, devised by Rudolf Flesch, is a distinctive measure of the readability of texts written in English. Simple to employ, the FRE formula factors in the average number of words per sentence and the average number of syllables per word to calculate a readability score ranging from 0 to 100 [7]. Indicative of the text's difficulty of comprehension, lower scores require higher levels of education to read and understand the text, while higher scores denote enhanced readability in the reading material [8].

The *Flesch-Kincaid Grade Level (FKG)* is a readability formula designed to facilitate an objective assessment of the reading difficulty of an English text. This formula combines the metrics of sentence length and word complexity to determine the educational grade level needed to comprehend a text [20]. The FKG formula yields a numeric score that corresponds to the approximate U.S. grade level necessary to understand the text effectively [11].

The *SMOG Index*, standing for Simple Measure of Gobbledygook, is a readability measure that evaluates the comprehension level of texts. The SMOG index focuses on quantifying the number of polysyllabic words present in the select sentences [13] by calculating a score indicative of the grade level required to comprehend the material. Although the index was standardized on 30-sentence samples, *textstat* can still produce a score for any texts with at least 3 sentences [1]. Within the healthcare and pharmaceutical industries, the SMOG index has been widely adopted for the linguistic evaluation of consumer-oriented healthcare materials.

The *Coleman-Liau Index (CLI)* is a readability test that estimates the understandability of a text. By factoring in the average number of letters per 100 words and the average number of sentences per 100 words, the CLI formula computes a score reflective of the U.S. grade level requisite for a proficient reading of the text [3].

The *Linsear Write (LINSEAR)* formula is a readability metric used to determine the readability level of materials written in English. Developed by Woodrow Linsear and Edward Thorndike, this formula calculates the U.S. grade level that indicates the estimated years of education recommended to read and understand a text comfortably. The calculation of readability using the LINSEAR formula involves tallying and assigning different weights to simple words containing one or two syllables and complex words encompassing three or more syllables [12].

The *Gunning Fog (GF)* is a commonly applied readability formula that specifies the degree of comprehensibility for English writings. Introduced by Robert Gunning, the GF formula estimates the reading level based on the average number of words per sentence and the percentage of polysyllabic words in a given text [18]. The reading level is a score indicative of the years of formal education a reader is required to understand the text on the initial reading. Users usually utilize the GF formula to validate that the text can be comprehended without difficulty by the intended audience.

The three remaining measures of text analysis are *Text Standard (textStd)*, *Lexicon Count (LexCount)* and *Difficult Words* [1]. TextStd is a readability consensus based upon all readability tests available in the *textstat* library on Python. It integrates all English readability formulas to return an estimated school grade level required to understand a written material. The next unique measure is LexCount, a valuable metric for gauging the richness in the vocabulary of a given text. LexCount calculates the total number of words present in a text and serves as a representation of the linguistic sophistication and lexical variation present within a written composition. Finally, the measure of Difficult Words quantifies the number of words with more than one syllable. The *textstat* library allows us to modify the minimum syllable count to any number more than 1 depending on our preference. We choose to set the parameter for a word to be counted as difficult at a minimum of 2 syllables since it would broaden our data range for the text analysis of programming problems we request ChatGPT to solve.

2.3 Online Competitive Programming Platforms

To test ChatGPT's ability to solve programming problems, we need a large amount of problems of varying difficulties and be able to measure and quantify these difficulties. There are many platforms, called online judge systems, that host programming problems and programming contests. Some of the more popular platforms include LeetCode and Kattis [19]. Earlier empirical work evaluating the effectiveness of automated coding support via GitHub' Copilot [14] showed that AI-based automated code suggestion can solve easy and medium difficulty categories of LeetCode while failing about 60% of hard questions. As ChatGPT is fairly new, many works analyzing ChatGPT are still under peer-review, but initial findings indicate that ChatGPT can effectively handle typical programming LeetCode problems [17]. A notable problem with using LeetCode for this type of study is the platform's own popularity. Solutions to LeetCode problems are widely available online. Furthermore, LeetCode problems are typically straight to point and do not contain extra details. This approach is different from traditional competitive programming problems where additional social and cultural background contexts are often included in problem statements for the purpose of distracting and confusing programmers. These types of problems are made available via Kattis, an online competitive programming platform that hosts archive of problems from numerous local, regional, and international competitive programming contests. Kattis' problems are categorized by their perceived level of difficulty ranging between 1.0 and 9.6., with 1.0 to 2.7 being classified as *easy*, 2.8 to 5.4 as *medium*, and 5.5 or higher as *hard* [10]. It should also be noted that solutions to Kattis' problems are not as popular on the Internet. It is perhaps because technical interview tends to use LeetCode type problems rather than the lengthy and verbose problems of Kattis. For our study, we used Kattis' problems.

3 Experimental Setup

This section describes how Kattis problems were selected, the process wherein their textual contents were cleaned up prior to submitting to ChatGPT for a solution, and how these solutions were evaluated. We also describe what final output data were collected and validated.

3.1 Problem selection and prompt generation

The Kattis problems for ChatGPT prompt generation are selected based on a number of criteria. Firstly, the problems were selected from a range of difficulty between 1.0 to 6.0. As shown in figure 1, these problems were distributed within five bins of difficulty (1.0 - 2.0, 2.0 - 3.0, 3.0 - 4.0, 4.0 - 5.0, and 5.0 - 6.0) with a minimum of 31 problems per bin. Secondly, the select problems contained only text. We used discretion in dismissing problems featuring anything (e.g., images) that could not be properly pasted into ChatGPT's prompt box. The problems' contexts span a range of subjects in mathematics, science, and language arts.

For each selected problem, we copied the entire text of the problem statement and any provided sample inputs and sample outputs from Kattis into the prompt box on ChatGPT. Prior to submitting the prompt, we also added a statement at the beginning to explicitly ask ChatGPT to solve the problem using Python and to provide justification for its solution. When copying problems from Kattis into ChatGPT, a number of mathematical expressions and notations cannot be directly included as they were embedded icons. These were to be filled out precisely by hand or by copying the Latex code into the prompt. All of the text on the ChatGPT prompt was stored in a *gpt_prompt* file, which was later used for our lexical analysis of the problems from Kattis.

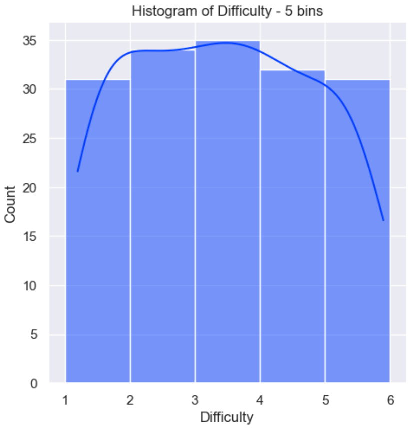


Figure 1: Histogram of Problems per Difficulty Range

3.2 Metric collections

After inputting the prompt, ChatGPT generated a Python code as a the solution. This solution was then used to submit for evaluation via Kattis. Upon submission, Kattis would run the code against a certain number of test cases to provide a judgement to the solution. The three statuses of judgement were *accepted*, *wrong answer*, and *error*. All correct test cases resulted in the *accepted* status for any solution. The *wrong answer* status would occur when any test case failed. We used the *error* status as a single category of judgements representing Kattis' *run-time error*, *time limit exceeded*, and *memory limit exceeded*. These mean that ChatGPT was unable to generate an optimized solution. Following a determination by Kattis of the status for every problem, we collected the problem difficulty, the status, and the score of test cases and stored these metrics as comments at the top a Python file that also contains ChatGPT's solution (*gpt.py*). The problem difficulty and status were particularly important to our metric analysis of ChatGPT's success in solving competitive programming problems in increased order of difficulty.

3.3 Output validation

As we evaluated ChatGPT's Python solutions by running the codes through Kattis' submission engine, we wanted to avoid scenarios where these solutions were correct but failed the tests due to incorrect formatting. For example, if an extra hyphen was added but was not supposed to be there, Kattis would mark the solution as wrong even if the values were correct. Examples of incorrect formatting include, but not limited to, unnecessary punctuation, rounding, and addition or omission of spaces. To account for this possibility, we manually inspected all ChatGPT solutions that received a *wrong answer* status and also did not pass any test cases. The solutions were tested on a local IDE environment using the sample inputs from Kattis. If a solution outputs incorrect results with with proper formatting, we can infer that the solution's algorithm is incorrect. However, if it gives correct results under incorrect formats, then we need to adjust the output code of the solution with proper formatting. In cases of a status *error* such as run-time error, time limit exception, or memory limit exception, we do not need to inspect the solution, even if it gets zero correct test cases. This is because we considered these particular errors to be direct results of incorrect implementation.

4 Results

4.1 Descriptive Statistics

Table 1 presented overall descriptive statistics of ChatGPT's performance, grouped into five difficulty ranges. These values were tabulated according to their solution status and calculations of the mean, minimum, maximum, and standard deviation values. The mean problem difficulty generally lies in the center of each of the five ranges, mainly because we considered all decimal values equally for every numeric difficulty in our selection of problems on Kattis and avoided bias towards sampling problems with the same exact difficulty more than a handful of times. Furthermore, the relatively low standard deviations indicate that the decimal values of the difficulty tend to be moderately clustered around the mean, and their dispersion is closely similar across all five ranges.

Figures 2 and 3 provided a visual intuition regarding the correlation between ChatGPT's performance and the problems' difficulty. The figures showed a negative correlation between the difficulty of the prompts and ChatGPT solutions with an *accepted* status. As the range of difficulty increases, the number of *accepted* solutions declines by a visible amount. The highest count of *accepted* solutions is observed in the two lowest difficulty ranges of 1.0 - 2.0 and 2.0 - 3.0, while the next two ranges, 3.0 - 4.0 and 4.0 - 5.0, account for a smaller proportion of correct answers. After trailing behind in the two lowest difficulty ranges, the count of *wrong answer* solutions began and continued exceeding the count of *accepted* solutions for all remaining ranges. Solutions with the *error* status is associated with an upward trend within the majority of difficulty ranges and have its highest count in the 4.0 - 5.0 range. There was no solution that passed all its test cases in the highest difficulty range of 5.0 - 6.0. Additionally, figure 2 is an illustration of the opposite distributions of the *wrong answer* and *accepted* statuses in terms of the problem difficulty. The *wrong answer* status mainly has a negatively skewed distribution, whilst a positively skewed distribution characterizes the *accepted* status. According to these distributions presented in figure 2, we can infer that the assigned difficulty of the prompts appears to be a reliable predictor of ChatGPT's success in generating an acceptable solution to a competitive programming problem.

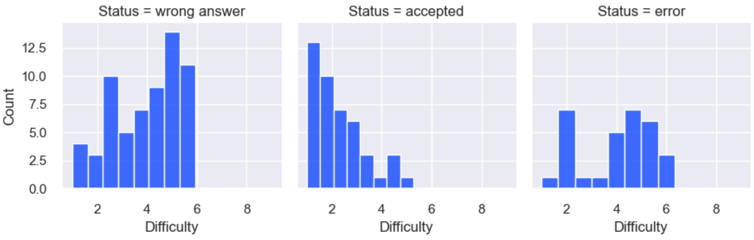


Figure 2: Distribution Graph of Status

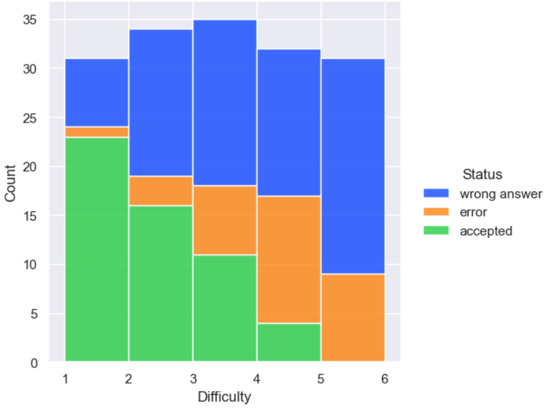


Figure 3: Histogram of Status per Difficulty Range

Table 1: Summary Statistics for Problem Difficulty of ChatGPT Prompts

Problem Difficulty: 1.0 - 2.0					
Status	**Count**	**Mean**	**Minimum**	**Maximum**	**Std. Dev.**
accepted	23	1.6	1.1	2.0	0.24
wrong answer	7	1.6	1.3	1.8	0.17
error	1	1.8	1.8	1.8	0
Total	31	1.6	1.1	2.0	0.24
Problem Difficulty: 2.0 - 3.0					
Status	**Count**	**Mean**	**Minimum**	**Maximum**	**Std. Dev.**
accepted	16	2.4	2.1	2.9	0.28
wrong answer	15	2.6	2.2	3.0	0.25
error	3	2.9	2.8	3.0	0.1
Total	34	2.5	2.1	3.0	0.28
Problem Difficulty: 3.0 - 4.0					
Status	**Count**	**Mean**	**Minimum**	**Maximum**	**Std. Dev.**
accepted	11	3.4	3.1	3.9	0.31
wrong answer	17	3.6	3.1	4.0	0.29
error	7	3.6	3.2	4.0	0.31
Total	35	3.5	3.1	4.0	0.3
Problem Difficulty: 4.0 - 5.0					
Status	**Count**	**Mean**	**Minimum**	**Maximum**	**Std. Dev.**
accepted	4	4.6	4.4	4.8	0.16
wrong answer	15	4.5	4.1	5.0	0.29
error	13	4.5	4.1	4.9	0.28
Total	32	4.5	4.1	5.0	0.28
Problem Difficulty: 5.0 - 6.0					
Status	**Count**	**Mean**	**Minimum**	**Maximum**	**Std. Dev.**
accepted	0	N/A	N/A	N/A	N/A
wrong answer	22	5.5	5.1	5.9	0.26
error	9	5.5	5.1	5.9	0.3
Total	31	5.5	5.1	5.9	0.27

4.2 Distribution of KS-Stat Test

To determine whether the differing nature of distributions of *accepted*, *wrong answer*, and *error* are by chance, we employed the Kolmogorov-Smirnov (KS) test in a pairwise fashion for these statuses. The results are presented in Table 2. The visual distribution graphs are shown in Figures 4 and 5. When comparing the distribution of difficulty in solutions that received *wrong answer* and *accepted* statuses, the test showed a p-value of $8.77 * 10^{-7}$, which presents statistically significant evidence that the distribution of difficulty level for *accepted* solutions are different from the distribution of difficulty level of *wrong answer* solutions. We also observed a similar result (p-value of $4.50 * 10^{-8}$) regarding the distribution of difficulty level between *error* and *accepted* solutions. On the other hand, there is no statistically significant evidence that solutions with *wrong answer* and *error* statuses came from different distribution. This provides support to our hypothesis that Kattis' problems' difficulty level can be used as a predictor to ChatGPT's ability to generate a correct and *accepted* solution. With this information, we examined whether the problems' textual contents can also be used as predictors. The KS test was used to observe if there were major differences in the distribution solution statuses basing on textual contents: ARI, DCR, FRE, LexCount, textStd, FKG, SMOG, CLI, Difficult Words, LINSEAR, and GF. The results of the tests were also presented in Table 2.

From the table, ARI, textStd, FKG, SMOG, CLI, and GF did not have any statistically significant results. This means that these measures of readability do not have a strong relationship with how difficult Kattis program are.

Comparing the distribution of *wrong answer* and *accepted* solutions using FRE showed statistically significant results. However, there was no statistically significant evidence when comparing the respective distribution pairs of *wrong answer* and *error* and *error* and *accepted.* Furthermore, the p-value when comparing *wrong answer* and *accepted* was not overwhelmingly significant (0.033). This possibly means that the readability measure used by FRE somewhat reflect the difficulty level of Kattis problem, but not strongly. Interestingly, *textstat*'s implementation of measuring difficult words (syllable-counting) gave similar statistical results as FRE's but with a much stronger p-value ($5.80 * 10^{-3}$) in the case of *wrong answer* and *accepted.* The difference between FRE and Difficult Word is that FRE also includes counts of words per sentence in its measurement.

Another readability measurements with only one significant p-value is LINSEAR, which showed a stronger statistical significance (0.02) when comparing the distribution between *error* and *accepted* but not for the other two distribution comparisons. Given that LINSEAR relies on the number of syllables to represent readability, this perhaps reflects the difference between fully understand a problem (hence *accepted*) versus partially understand a problem (missing the nuisances, hence *error*: memory limit, time limit, and run time).

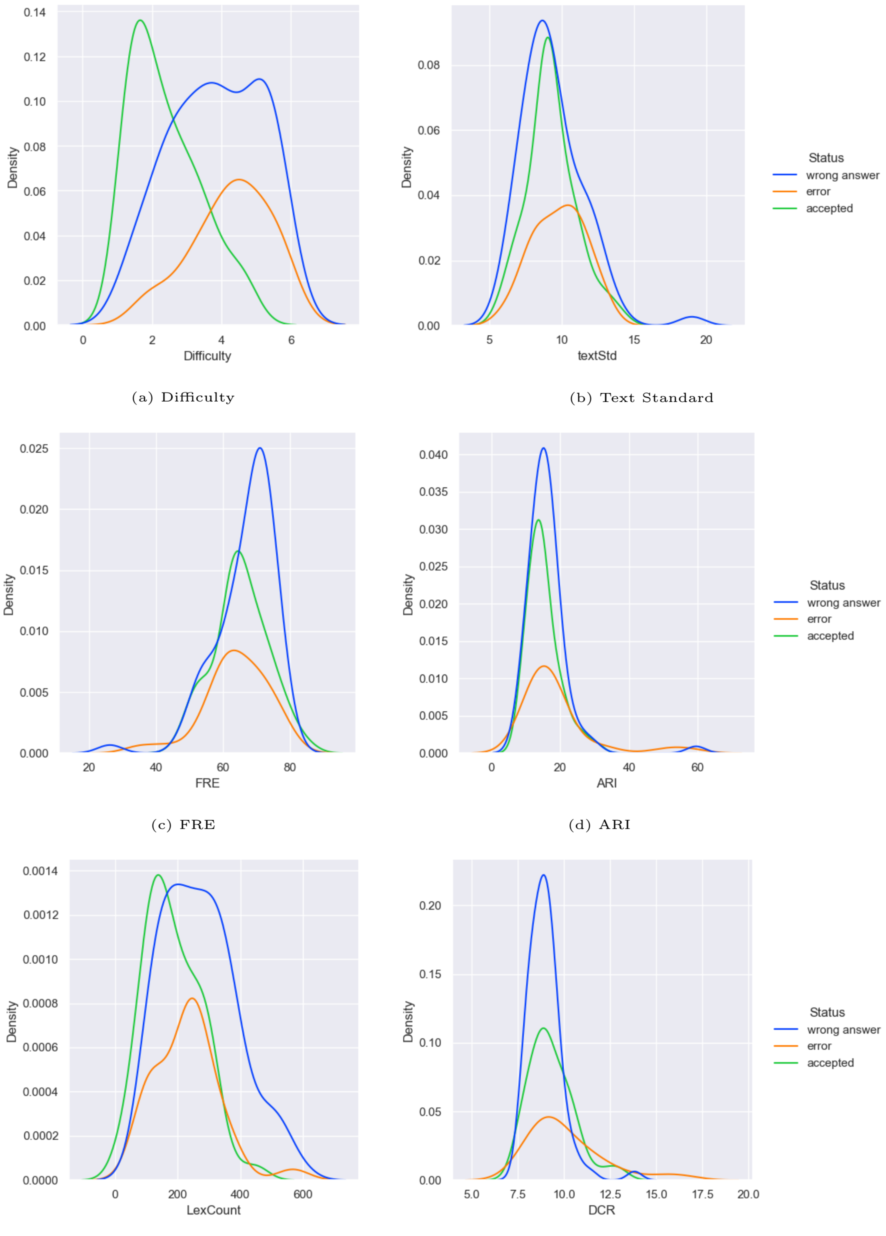
In the case of DCR, for *wrong answer* and *accepted*, we observed a statistically significant result (p-value of 0.045). This was also the case for the distribution of *error* and *accepted* solutions (p-value of 0.035) as did wrong *wrong answer* and *error* (p-value of $2.50 * 10^{-5}$). While this is indicative of a strong relationship between a problem's DCR score and the final status of the problem, the p-values for the distribution pairs of *wrong answer* and *accepted* and *error* and *accepted* were also barely clear the 0.05 threshold for statistical significane. This remains a promising lead, as it turns out that DCR is considered able to measure context of the writing itself [1].

Among all measures, LexCount is the only that had statistical results most similar to difficulty level. When comparing the distribution between *wrong answer* and *accepted* solutions, we received a p-value of $7.49 * 10^{-4}$. This is also the case between *error* and *accepted* solutions which (p-value of 0.035), albeit not as strong. The KS test for *wrong answer* and *error* solutions did not have a statistically significant result but the p-value of 0.056 is very close to becoming one. This implies a potential strong similarity in being able to use Lexicon Count or difficulty level in predicting solution status. This is perhaps due to the fact that ChatGPT does not process words like a human. It does not care about sentences and and how many there are, and is more concerned about the total number of words in the text and to some degree, the number of syllables as representative of readability.

Table 2: Summary Statistics for Kolmogorov-Smirnov Test Results using Different Distributions

Distributions	*wrong answer* vs *accepted*	*wrong answer* vs *error*	*error* vs *accepted*
Difficulty	$8.77 * 10^{-7}$ ***	0.315	$4.50 * 10^{-8}$ ***
ARI	0.613	0.240	0.359
DCR	0.045 *	$2.50 * 10^{-5}$ ***	0.035 *
FRE	0.033 *	0.106	0.359
LexCount	$7.49 * 10^{-4}$ ***	0.056	0.035 *
textStd	0.625	0.302	0.215
FKG	0.824	0.274	0.209
SMOG	0.832	0.248	0.565
CLI	0.064	0.115	0.285
Difficult Words	$5.80 * 10^{-3}$ **	0.553	0.115
LINSEAR	0.318	0.216	0.020 *
GF	0.937	0.150	0.195

Figure 4: Graphs of Distributions Separated by Status



(a) Difficulty

(b) Text Standard

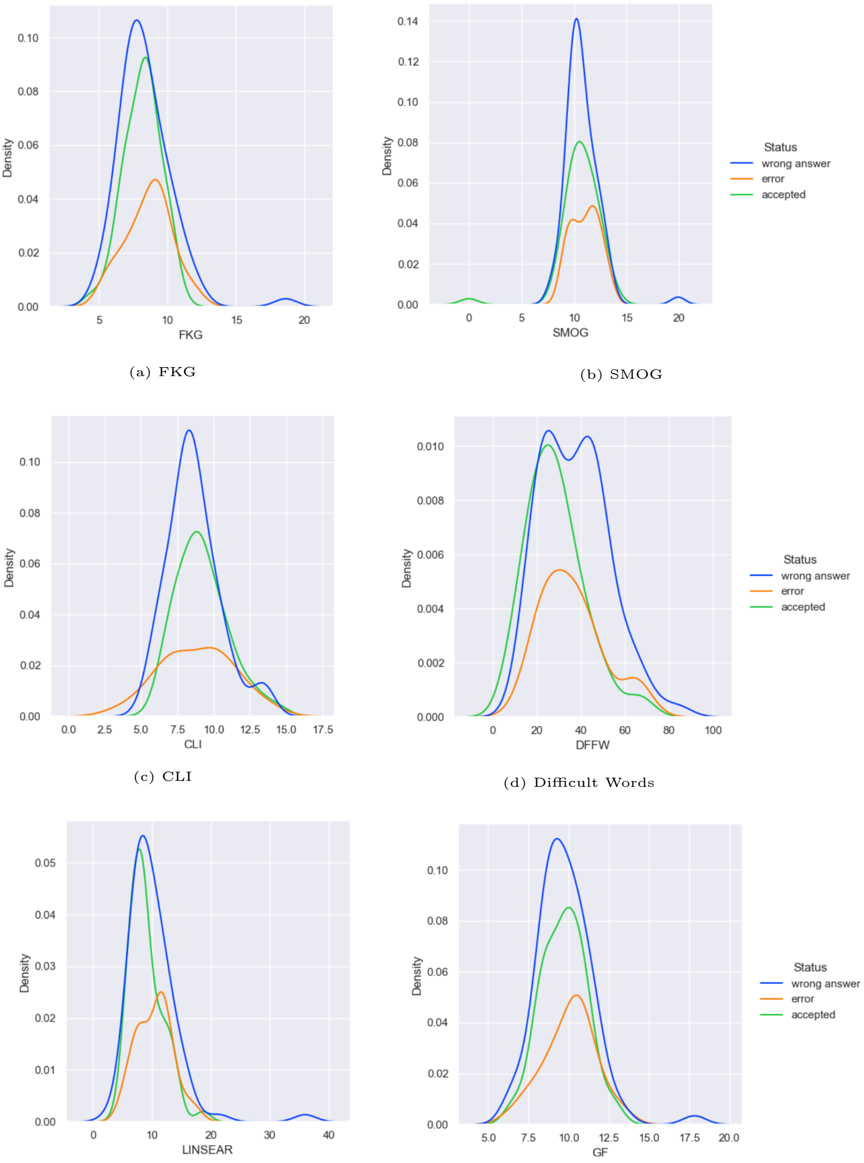
(c) FRE

(d) ARI

(e) Lexicon Count

(f) DCR

Figure 5: Graphs of Distributions Separated by Status Cont.



(a) FKG

(b) SMOG

(c) CLI

(d) Difficult Words

(e) Linsear

(f) GF

5 Conclusion

From the data we have collected, we found many interesting observations on the impact of the readability and textual components have on ChatGPT's solution status. Kattis difficulty level and Lexicon Count were seen to have the highest impacts in correlating between different statuses of a solution generated by ChatGPT. Several other readability measurements showed some relationship but not as comprehensive as the two above. This indicates that it is possible to generate problems whose descriptions are complex enough for humans to understand but not for ChatGPT.

The preliminary results in this paper provided insights to future study regarding how ChatGPT responds to the textual components of a problem. This includes but not limited to.

- Continue collecting, cleaning, and adding more problems at different difficulty level to have more data regarding specific error types.
- Analyze ChatGPT's justification texts to understand how the solution was constructed.
- Perform in-depth analysis on various readability measures that showed statistical significant results here. Develop new comprehensive measures that could better represent technical difficulty of programming problems.

The data collected for this work is made publicly available at [redacted].

References

[1] Shivam Bansal and Chaitanya Aggarwal. *Textstat*. Mar. 2022. URL: `pypi.org/project/textstat/`.

[2] *ChatGPT*. https://openai.com/blog/chatgpt. 2023.

[3] Meri Coleman and Ta Lin Liau. "A computer readability formula designed for machine scoring." In: *Journal of Applied Psychology* 60.2 (1975), p. 283.

[4] Edgar Dale and Jeanne S Chall. "A formula for predicting readability: Instructions". In: *Educational research bulletin* (1948), pp. 37–54.

[5] Eva AM van Dis et al. "ChatGPT: five priorities for research". In: *Nature* 614.7947 (2023), pp. 224–226.

[6] Tyna Eloundou et al. "Gpts are gpts: An early look at the labor market impact potential of large language models". In: *arXiv preprint arXiv:2303.10130* (2023).

[7] Rudolph Flesch. "A new readability yardstick." In: *Journal of applied psychology* 32.3 (1948), p. 221.

[8] Daniel J Gallagher and G Rodney Thompson. "A readability analysis of selected introductory economics textbooks". In: *The Journal of Economic Education* 12.2 (1981), pp. 60–63.

[9] Sajed Jalil et al. "Chatgpt and software testing education: Promises & perils". In: *arXiv preprint arXiv:2302.03287* (2023), pp. 1–2.

[10] *Kattis Problem Archive.* https://open.kattis.com/. 2023.

[11] J Peter Kincaid et al. *Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel.* Tech. rep. Naval Technical Training Command Millington TN Research Branch, 1975.

[12] George R Klare. "Assessing readability". In: *Reading research quarterly* (1974), pp. 62–102.

[13] G Harry Mc Laughlin. "SMOG grading-a new readability formula". In: *Journal of reading* 12.8 (1969), pp. 639–646.

[14] Nhan Nguyen and Sarah Nadi. "An empirical evaluation of GitHub copilot's code suggestions". In: *Proceedings of the 19th International Conference on Mining Software Repositories.* 2022, pp. 1–5.

[15] M Ruby. "How chatgpt works: The model behind the bot". In: *Towards Data Science* (2023).

[16] RJ Senter and Edgar A Smith. *Automated readability index.* Tech. rep. Cincinnati Univ OH, 1967, pp. 5–9.

[17] Haoye Tian et al. "Is ChatGPT the Ultimate Programming Assistant–How far is it?" In: *arXiv preprint arXiv:2304.11938* (2023).

[18] Tiffany M Walsh and Teresa A Volsko. "Readability assessment of internet-based consumer health information". In: *Respiratory care* 53.10 (2008), pp. 1310–1315.

[19] Szymon Wasik et al. "A survey on online judge systems and their applications". In: *ACM Computing Surveys (CSUR)* 51.1 (2018), pp. 1–34.

[20] Mostafa Zamanian and Pooneh Heydari. "Readability of Texts: State of the Art." In: *Theory & Practice in Language Studies* 2.1 (2012).

Developing Data Protection and Recovery Plan for Healthcare IoT Domain*

Syed Rizvi, Jessica Ayres, Jessie Pensyl, Mark Ihnat
Department of Information Sciences and Technology
Pennsylvania State University, Altoona, PA
{srizvi, jfa5460, jkp5537, mzi70}@psu.edu

Abstract

The healthcare sector serves as an indispensable component within our societal framework. Within this realm, the Internet of Things (IoT) plays a vital role in facilitating the rapid transmission of voluminous data and crucial patient information. Organizations have conscientiously implemented risk management practices to assess potential risks inherent and diligently scrutinized various solutions to mitigate these risks effectively. This research endeavors to augment our analysis by integrating the Business Impact Analysis (BIA) framework, enabling us to comprehensively determine the requisite measures that healthcare domain should undertake when confronted with a disaster scenario. By leveraging BIA, we aim to identify strategies to minimize damage, expedite recovery, and promptly restore normal operational capabilities. Furthermore, it is crucial to highlight that relying solely on compliance measures like HIPAA may not be sufficient to ensure the security of patient information and overall patient health in the context of IoT devices used in doctor's offices and hospital environments. Therefore, this research emphasizes the need for regular implementation of additional precautions such as the Business Impact Analysis (BIA) framework and robust risk management practices. Neglecting these proactive measures puts patient information at risk and can potentially compromise patient health. The paper underscores the importance of going beyond compliance regulations to address the unique challenges and risks associated with IoT devices in healthcare settings.

1 Introduction

IoT devices have become pervasive within the healthcare sector, offering a multitude of advantages that contribute to their remarkable capabilities. Their applications encompass diverse areas such as remote patient monitoring, disease and medication management, operational efficiency, and emergency response and safety. Remote patient monitoring, for instance, enables healthcare professionals to remotely track vital indicators like heart rate, blood pressure, and sleep patterns. Disease management entails the collection of patient data outside the confines of hospitals or clinics, while medication management ensures timely reminders for medication intake and alerts for refills. The utilization of IoT devices yields numerous benefits, including enhanced patient outcomes, cost reduction, and patient empowerment. Through proactive monitoring, potential health issues can be identified at an early stage, enabling timely intervention by medical practitioners before the conditions deteriorate. Simultaneously, such monitoring mitigates costs for all stakeholders by curbing unnecessary visits and readmissions. Moreover, patient empowerment is fostered as individuals gain greater control over their health, actively participating in their care and making well-informed decisions.

However, it is imperative to acknowledge that Healthcare IoT is not immune to flaws and vulnerabilities, as is the case with any Internet-connected system. Among the primary vulnerabilities lies the inadequate security measures implemented in many devices, characterized by default passwords and the absence of encryption protocols. These security shortcomings render the devices susceptible to exploitation, granting unauthorized access to sensitive personal data. The prevalence of malware and ransomware attacks further compounds the issue. While the number of ransomware attacks has witnessed a decline since the previous year, the monetary demands imposed by attackers have escalated. The unavailability of patient records significantly hampers the efficient functioning of hospitals, placing individuals at risk of having their private information compromised and disseminated without consent. Compounding the challenge is the infrequent updating of healthcare IoT devices to rectify known vulnerabilities, thus rendering them susceptible to exploitation and other menacing.

Additionally, as is the case in any organization, employees represent a potential weak point in the security chain. Insufficient training regarding best practices, such as identifying phishing emails or inadvertently exposing critical information, increases the likelihood of human errors and subsequently exacerbates the security risks. Insider threats exacerbate this predicament; disgruntled employees may misuse equipment for personal gain. Moreover, the prevalence of Denial-of-Service (DoS) attacks poses a significant concern, as they can incapacitate services and equipment, thereby jeopardizing critical procedures such as surgeries.

As previously discussed, the implementation of IoT in the healthcare sector yields a plethora of applications. One notable application involves the monitoring of vital health indicators, including blood pressure, glucose levels, and body temperature [6]. Such monitoring proves invaluable in assisting both physicians and patients in treatment plans and medication management while also enabling cost savings and facilitating early detection of potential health issues. In efforts to further reduce costs, experts predict that the integration of inexpensive tags onto essential items will soon be feasible, with estimates as low as 10 cents per tag [5]. These RFID tags will effectively track the location of equipment and other vital tools, optimizing operational efficiency. Despite these positive developments, numerous challenges persist within the realm of healthcare IoT devices. For instance, when employing monitoring systems that rely on pattern recognition, the utilization of outdated datasets can pose a significant challenge, potentially leading to erroneous decisions based on obsolete results [4]. Other challenges encompass processing vast volumes of data at high speeds, ensuring sufficient bandwidth capacity to handle such data along with the required infrastructure, and mitigating the risks of security breaches and potential hacking incidents, as the interconnected nature of these devices renders them vulnerable [1]. Recognizing these vulnerabilities and threats, IoT is also being harnessed to bolster disaster recovery efforts, extending beyond the confines of hospitals. Leveraging drone technology, for instance, facilitates the swift location of accident victims. Drones equipped with HSV Color Detection and HOG Feature Extraction capabilities can efficiently scan an area and identify individuals in need of assistance [2]. This technological advancement holds the potential to save lives by expediting rescue operations. Additionally, alongside the use of drones, the formulation of a robust incident response plan becomes indispensable, as it effectively addresses various security challenges, encompassing system availability, authentication and access control, data integrity, data loss prevention, and more [3].

The establishment of robust data recovery and contingency planning protocols is of paramount importance for several compelling reasons. Foremost, these measures ensure the preservation and safeguarding of data, thereby significantly reducing the likelihood of complete data loss. Considering the substantial volume of sensitive information generated and stored by these devices, including medical records, diagnostic data, and treatment records accessible round-the-clock, a comprehensive data recovery plan serves as a crucial safeguard against potential hardware failures, cyberattacks, or natural calamities. The loss of such data, for any reason, can prove highly detrimental to both the healthcare provider and the well-being of their patients. Additionally, the safety of patients themselves is a matter of grave concern. As numerous IoT devices are deployed for health monitoring, and the management of medication and treatments, any malfunction in these devices can pose a severe risk to patient safety.

Moreover, due to the stringent regulations in place to protect the confidentiality of patient data, such as the Health Insurance Portability and Accountability Act (HIPAA), organizations that fail to adequately safeguard patient information may face legal repercussions and endure substantial damage to their reputation. An effective contingency plan not only enables healthcare providers to meet regulatory standards but also ensures a prompt and efficient response to security incidents. Data recovery plans encompass proactive measures aimed at preventing and mitigating risks, as well as ensuring the secure retrieval of data in the event of an unforeseen event. Given the paramount significance of privacy and security in this sector, organizations that demonstrate a well-designed contingency plan exhibit their unwavering commitment to data protection and their ability to manage emergency situations effectively. By doing so, they enhance stakeholder trust and instill confidence in consumers regarding their data protection practices.

This paper aims to comprehensively present the intricate dynamics of IoT devices and their multifaceted interactions within the healthcare domain.

Through the utilization of Tables and Figures, a comprehensive depiction is provided to elucidate the network structures, organizational assets, various breach typologies, and comprehensive plans encompassing data protection and recovery strategies, among other pertinent aspects. The paper commences with a compelling case study, which serves as a foundation for subsequent discussions, followed by a comprehensive Risk Assessment that delves into the intricacies of the Risk Management Framework, accompanied by pertinent recommendations to guide organizations in their risk mitigation efforts. Subsequently, the BIA is meticulously expounded upon, emphasizing the imperative nature of prompt and decisive actions required to facilitate effective recovery in the aftermath of an attack. Lastly, comprehensive plans are formulated for Data Recovery and Contingency, offering detailed step-by-step instructions to facilitate efficient recovery and the resumption of normal operations following a security breach.

2 Case Study

In the course of this study, a hypothetical IoT infrastructure was meticulously constructed, as visually represented in Figure 1. The healthcare domain was effectively partitioned into distinct groups, each comprising a diverse array of devices responsible for critical day-to-day operations. Three primary components were developed, namely the healthcare public cloud, the examination room network, and the operating room network. A connection to external entities originating from the public cloud was established, facilitating seamless integration with third-party stakeholders. Additionally, a selection of end devices, encompassing wireless devices such as smartphones and laptops, were strategically incorporated into the overall framework. The healthcare domain was meticulously constructed, and an extensive risk assessment was conducted, with a specific focus on evaluating the IoT infrastructure. To effectively manage these risks, a comprehensive range of risk management activities was executed, encompassing assessment, identification, and evaluation of critical assets, followed by the prioritization of risks based on their significance. Subsequently, an in-depth analysis was conducted to ascertain the most suitable risk mitigation strategies.

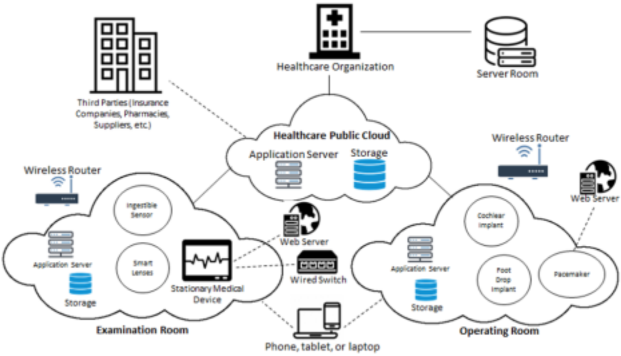


Figure 1: IoT Healthcare Diagram

Notably, the primary risk area identified pertains to potential system shutdowns or viral attacks, as such disruptions severely impede the execution of daily operations. Given the critical nature of the healthcare domain, which encompasses intricate details concerning medications and patient data, ensuring the secure and uninterrupted functionality of our devices becomes paramount. Safeguarding the confidentiality of patient information emerges as one of the foremost concerns, as any breach in this regard not only poses financial repercussions but also inflicts significant damage to the organization's reputation.

Having concluded our comprehensive risk assessment and management endeavors, we can now delve into the crucial BIA phase. The overarching objective of formulating a robust BIA is to establish an operational framework that guarantees the seamless functioning of our healthcare domain while simultaneously devising a meticulously crafted contingency plan to address potential breaches or disasters. By leveraging the BIA methodology, we will effectively identify, evaluate, and prioritize our critical functions that necessitate stringent safeguarding measures. Consequently, this analytical approach will empower us to ascertain the precise actions our organization must undertake in the event of an unforeseen incident, thereby facilitating the minimization of damage, expeditious system recovery, and the swift restoration of normal operational functionality.

Drawing upon the risk management methodologies outlined in the preceding paper, we have successfully identified and prioritized the risks inherent within our healthcare domain. This foundational undertaking has significantly contributed to the formulation of a comprehensive BIA, as it provides us with a solid groundwork from which to determine our most critical functions and devise appropriate strategies to mitigate their impact in the event of a disaster. Resilience and swift recovery from any potential attacks remain paramount objectives for our organization. Through the meticulous application of BIA, we will systematically deconstruct each step, comprehensively assessing the tolerance levels specific to our healthcare domain. This encompassing analysis encompasses various factors, including the acceptable threshold for data loss, allowable system downtime duration, and estimated recovery timelines required to reinstate full operational capabilities. The identification of resource requirements will be meticulously undertaken, allowing us to ascertain the essential resources needed to sustain business continuity even in the aftermath of an attack. Lastly, our BIA endeavors will culminate in determining the prioritization of system resource recovery, thus establishing a clear roadmap to restore critical functionalities efficiently.

3 Risk Management Framework

The initial phase of conducting a comprehensive risk assessment entails the meticulous identification of all organizational assets, coupled with a comprehensive evaluation of the corresponding threats that pose potential risks to these assets. For visual clarity, Figure 2 displays a graphic representation of our Risk Management Framework.

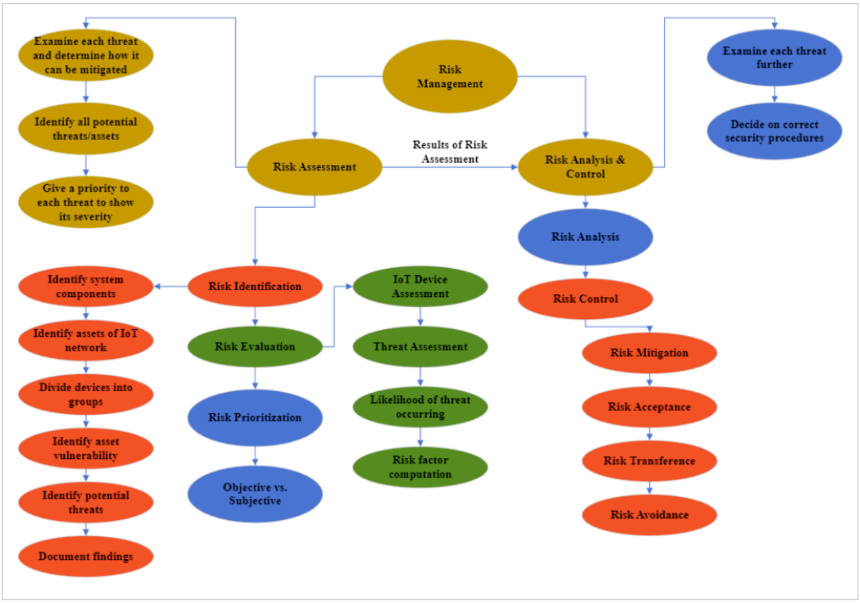
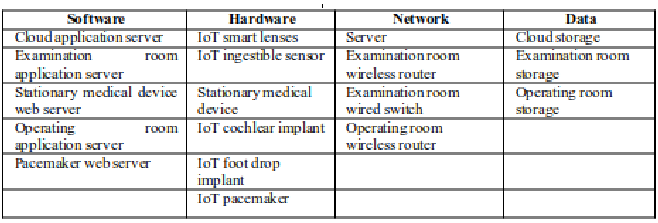

Examine each threat and determine how it can be mitigated
Identify all potential threats/assets
Give a priority to each threat to show its severity
Risk Management
Risk Assessment
Results of Risk Assessment
Risk Analysis & Control
Examine each threat further
Decide on correct security procedures
Risk Analysis
Risk Control
Risk Mitigation
Risk Acceptance
Risk Transference
Risk Avoidance
Risk Identification
Identify system components
Identify assets of IoT network
Divide devices into groups
Identify asset vulnerability
Identify potential threats
Document findings
Risk Evaluation
Risk Prioritization
Objective vs. Subjective
IoT Device Assessment
Threat Assessment
Likelihood of threat occurring
Risk factor computation

Figure 2: Risk Management Framework for IoT Network

Table 1: List of Assets

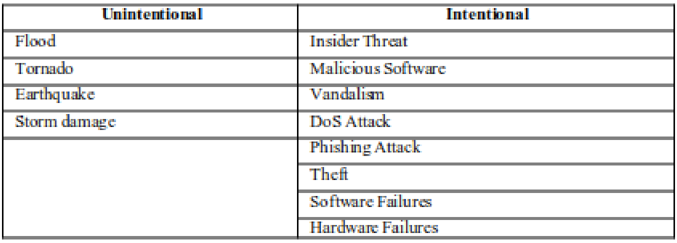


Software	Hardware	Network	Data
Cloud application server	IoT smart lenses	Server	Cloud storage
Examination room application server	IoT ingestible sensor	Examination room wireless router	Examination room storage
Stationary medical device web server	Stationary medical device	Examination room wired switch	Operating room storage
Operating room application server	IoT cochlear implant	Operating room wireless router	
Pacemaker web server	IoT foot drop implant		
	IoT pacemaker		

Given our specific focus on the IoT infrastructure, it becomes imperative to consider all interconnected networks alongside the associated IoT devices. Accordingly, our examination commences by meticulously identifying and documenting the organizational assets, followed by an exhaustive assessment of the corresponding threats. Within our hypothetical healthcare organization, a multitude of assets necessitate identification. To facilitate this process, a succinct bullet-point format is employed to document all assets, subsequently enabling us to prioritize them based on their relative significance. The assets within our organization have been systematically categorized into distinct groups, as illustrated in Table 1, to facilitate the identification of group-specific threats.

Our asset inventory comprises four main categories: software, hardware, network, and data. Under the software category, we have included all application servers across our organization's three branches, as well as the web servers utilized by certain hardware components. The hardware category encompasses our extensive array of IoT devices, in addition to a stationary medical device. The network section includes our primary server, two routers, and a wired switch. Lastly, the data group encompasses all storage resources integral to our operations. Alongside asset identification, it is crucial to assess the vulnerabilities our assets may possess. These vulnerabilities encompass a range of potential risks, such as unpatched software exploits, unchanged default passwords, inadequate access control, undocumented software instances, insufficient control over physical access to premises, and numerous others. Subsequently, the next phase of our risk assessment entails identifying all threats that pose risks to our organizational assets. These threats can manifest in various forms, including natural disasters or intentional malicious actions by external attackers. A comprehensive listing of select threats pertinent to our organization can be found in Table 2 below. Once all assets and threats have been identified, the subsequent phase involves their evaluation to establish priorities. Given the significant presence of IoT devices within our organization, a more comprehensive assessment of these devices becomes imperative. This entails acquiring detailed knowledge of the software utilized by our IoT devices, as well as comprehending the intricacies of data collection processes associated with them.

Table 2: Unintentional and Intentional Breaches



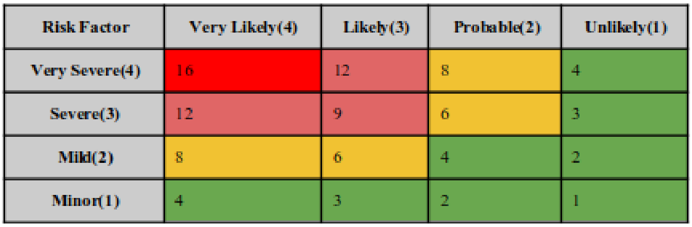
Unintentional	Intentional
Flood	Insider Threat
Tornado	Malicious Software
Earthquake	Vandalism
Storm damage	DoS Attack
	Phishing Attack
	Theft
	Software Failures
	Hardware Failures

To facilitate prioritization, we will employ a scale that takes into account the prevalence of threats and the value of assets. Our risk factor calculation will be based on a scale ranging from one to sixteen, wherein sixteen denotes the highest level of criticality, and one denotes the lowest. This scale, exemplified in Table 3, will be utilized for auditing our IoT devices and determining their respective risk factors. The final stage in the risk assessment process involves prioritization. Utilizing the risk computation table mentioned earlier, we will discern the risk factors that demand the highest level of attention, taking into consideration both the probability and the magnitude of the associated threats. The magnitude of a threat will be assessed based on the monetary value of the targeted asset. It is important to note that the severity evaluation is subjective, as it relies on the available knowledge and information at hand.

3.1 Recommendations

The subsequent step in the risk management process entails mitigating the identified risks. We propose several recommendations to enhance the security measures of the healthcare organization. Foremost among these is the implementation of antivirus software on all devices utilized within the facility. Given the substantial number of devices in operation, the risk of encountering viruses and similar threats is significant.

Table 3: Critical Threats



Risk Factor	Very Likely(4)	Likely(3)	Probable(2)	Unlikely(1)
Very Severe(4)	16	12	8	4
Severe(3)	12	9	6	3
Mild(2)	8	6	4	2
Minor(1)	4	3	2	1

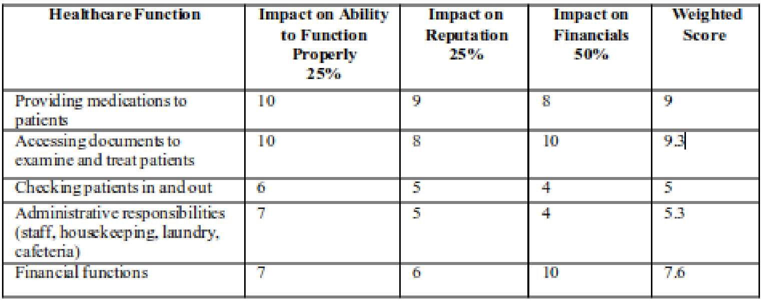
An infection affecting any of these devices could have severe consequences for the organization. The interconnected databases contain a wealth of sensitive information, and the compromise of even a single device could give rise to substantial concerns. Not only would such an incident jeopardize the integrity of the stored data, but it would also undermine the organization's reputation. The confidentiality of client information is of paramount importance to stakeholders and consumers alike. Without their trust and support, the organization's ability to function effectively would be severely hampered.

In addition to implementing antivirus software, we propose the comprehensive training of all employees in fundamental computer safety practices. Human error stands as a prominent catalyst for information breaches within organizations. Ensuring that employees receive adequate training can significantly minimize the risk of inadvertent mistakes leading to security breaches. Simple yet crucial training initiatives, such as recognizing and handling spam emails, regularly changing passwords, employing unique passwords for each account, and other fundamental techniques, can substantially curtail the potential for attacks. Moreover, addressing physical security risks is imperative, particularly concerning the server room, which houses the organization's crucial data. Stringent protective measures, including physical locks, restricted access granted only to authorized personnel, and adherence to standardized room specifications to optimize server functionality, must be implemented to safeguard this critical area. Even the slightest vulnerability in the server room presents an opportunity for an attack that could result in the complete loss of organizational data. We also recommend risk transference as a means of reducing overall risk. This entails transferring the risk to a willing third party, typically through insurance coverage. By obtaining suitable insurance, the organization can alleviate financial burdens in the event of an attack, potentially saving millions. However, it is important to note that insurance coverage does not address the impact on the organization's reputation, as public perception and their assessment of the situation remain beyond anyone's direct control.

4 Outcome of BIA

In this section, we will comprehensively explore each step of the BIA process. The initial step entails the analysis of business functions, wherein we delve into the overarching objectives of the organization. Specifically, within the context of a healthcare domain, our primary aim is to ensure the provision of optimal treatment to all patients.

Table 4: Function Weighted Prioritization Table



Healthcare Function	Impact on Ability to Function Properly 25%	Impact on Reputation 25%	Impact on Financials 50%	Weighted Score
Providing medications to patients	10	9	8	9
Accessing documents to examine and treat patients	10	8	10	9.3
Checking patients in and out	6	5	4	5
Administrative responsibilities (staff, housekeeping, laundry, cafeteria)	7	5	4	5.3
Financial functions	7	6	10	7.6

Consequently, in this step, we must ascertain which functions are critical for sustaining this overarching objective. To facilitate this aspect of the BIA, we have devised a weighted analysis table, which allows us to systematically list and evaluate various routine functions. Subsequently, through the process of prioritization, we derive a ranked catalog of our essential business processes.

In Table 4, we have included a selection of pivotal tasks that our domain is responsible for executing. This diagram offers valuable insights into the significance of two key functions: ensuring the accurate administration of medications to our patients and facilitating seamless access to pertinent documents for comprehensive examination and treatment purposes. Having identified these critical functions, we can now transition to the subsequent stage of the BIA process. In this phase, we will delve into the meticulous breakdown of recovery criticality, encompassing the expeditious restoration of our essential functions in the event of a disaster. To accomplish this, we must gauge our organization's tolerance levels by determining essential metrics such as the recovery point objective (RPO), recovery time objective (RTO), work recovery time (WRT), and maximum tolerable downtime (MTD).

Initially, we will proceed with the determination of our RPO. This entails evaluating the extent of data loss that can be deemed acceptable within our operations. As an illustration, within a healthcare facility, the data undergoes constant modifications, with updates occurring at a rapid pace. These updates encompass a diverse range of information, encompassing critical details such as patients' conditions, prescribed medications and dosages, patient locations or transfers, and even minute particulars, including dietary restrictions and meal schedules. Given the crucial nature of this data, our RPO must be established within a timeframe of 0 to 1 hour. Healthcare domains inherently handle substantial volumes of data traffic, necessitating a notably low RPO.

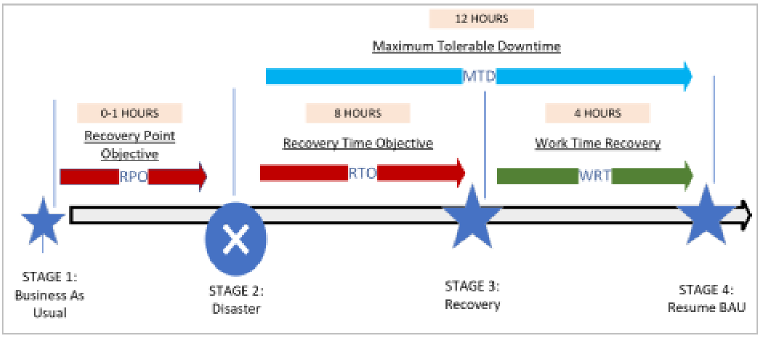
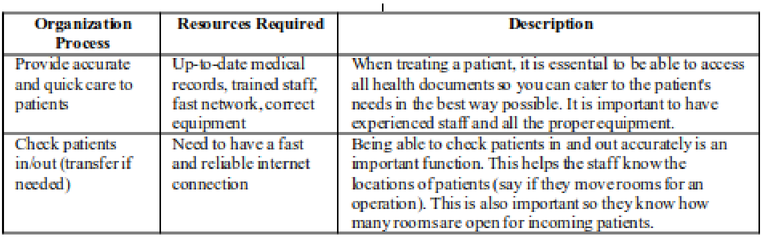


Figure 3: Timeline Diagram

Subsequently, we proceed to establish our RTO. The RTO delineates the maximum permissible duration during which a system can remain inaccessible before impeding the functionality of other essential resources. Considering the multifaceted aspects involved in patient care and the continuous flow of data, we estimate that the RTO must not exceed 8 hours to maintain seamless operations.

Subsequently, we will proceed to determine the WRT, constituting the maximum permissible duration required to restore the systems effectively. This entails comprehensive testing and verification procedures to ascertain the proper functioning of the systems and the successful recovery of data. Within our hypothetical healthcare domain, envision a dedicated team diligently working towards expeditious system restoration. Consequently, the WRT is expected to encompass a relatively concise timeframe, estimated not to exceed 4 hours. Finally, we arrive at the MTD, which is the aggregate of the RTO and the WRT. The MTD denotes the total duration during which a business can experience disruption without incurring detrimental consequences. By adding our RTO of 8 hours to the WRT of 4 hours, our computed MTD equates to 12 hours. This temporal threshold assumes crucial significance in the formulation of our continuity policies. It ensures that our recovery procedures align with the acceptable downtime threshold (in our specific case, 12 hours). For a comprehensive depiction of this entire process, please refer to Figure 3, presented below. The subsequent stage in the BIA entails the identification of requisite resources. To facilitate this process, we have constructed Table 5, which briefly delineates the resources necessary for system recovery in the event of a calamity. The final stage of the BIA involves the identification of resource recovery priorities. This critical step assists us in determining the order in which resources should be restored. As highlighted in previous sections, the seamless access to patient information is deemed paramount within our healthcare domain.

Table 5: Needed Resources in Disaster



Organization Process	Resources Required	Description
Provide accurate and quick care to patients	Up-to-date medical records, trained staff, fast network, correct equipment	When treating a patient, it is essential to be able to access all health documents so you can cater to the patient's needs in the best way possible. It is important to have experienced staff and all the proper equipment.
Check patients in/out (transfer if needed)	Need to have a fast and reliable internet connection	Being able to check patients in and out accurately is an important function. This helps the staff know the locations of patients (say if they move rooms for an operation). This is also important so they know how many roomsare open for incoming patients.

5 Contingency Policies for Healthcare Domain

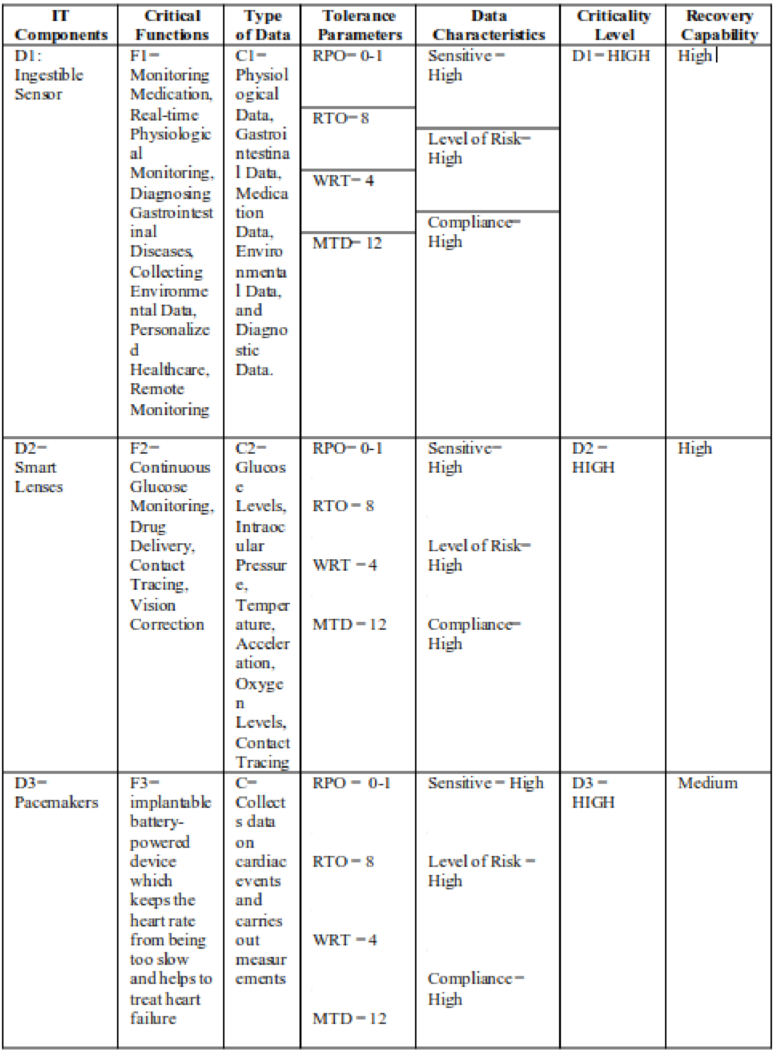
Following the development of the BIA, we have formulated recommendations to establish a robust contingency policy. This contingency policy incorporates key metrics such as RPO, RTO, WRT, and MTD specific to the healthcare domain. Its primary objective is to safeguard patient data. To achieve this, several security policies must be implemented within the healthcare domain. These include safeguarding critical hardware by implementing duplication or maintaining backup copies.

Additionally, regular updates and backups of the operating system and software applications are crucial to prevent data loss. During downtime, policies and procedures should be implemented to verify patient identity both pre and post-breaches or disruptions. Providing paper forms as substitutes for digital forms during such periods can help ensure organizational functionality. Also, it is imperative to equip employees with appropriate training to effectively manage downtimes, recovery periods, and common security threats.

In the event of a security breach, it is imperative to communicate promptly with both employees and the wider public to ensure individuals are aware of potentially compromised information. To mitigate the risk of future breaches, healthcare organizations should regularly test their systems and implement robust monitoring measures to prevent potential downtimes proactively. By implementing these recommendations, the contingency policy is expected to enhance the operational system of the healthcare organization and effectively minimize any potential downtime.

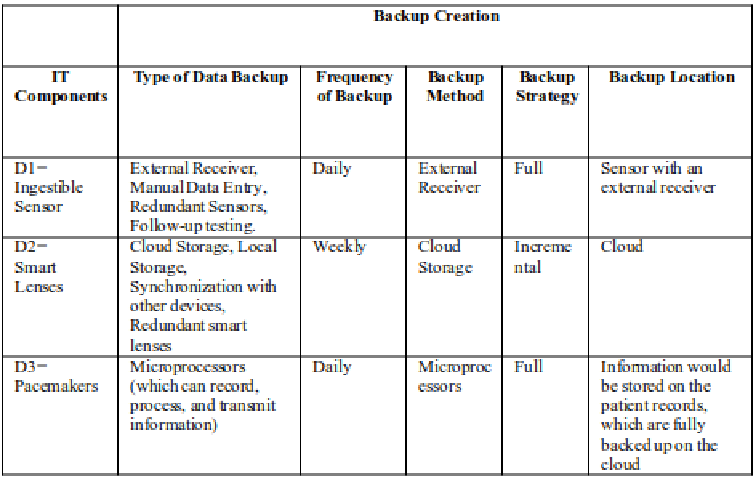
Table 6 presents the IoT Device Data Protection and Recovery Analysis, encompassing crucial information such as the name of the IoT device, its associated critical functions, the type of data involved, tolerance parameters, data characteristics, criticality level, and recovery capability. Moving on to Table 7, we provide a comprehensive plan for IoT device data protection and recovery, which outlines the backup methodology employed for each device. This includes the frequency of backups, the selected backup method, and the corresponding strategy. Additionally, the designated backup location for each device is specified.

Table 6: IoT Device Data Protection and Recovery Analysis



IT Components	Critical Functions	Type of Data	Tolerance Parameters	Data Characteristics	Criticality Level	Recovery Capability
D1: Ingestible Sensor	F1= Monitoring Medication, Real-time Physiological Monitoring, Diagnosing Gastrointestinal Diseases, Collecting Environmental Data, Personalized Healthcare, Remote Monitoring	C1= Physiological Data, Gastrointestinal Data, Medication Data, Environmental Data, and Diagnostic Data.	RPO= 0-1 RTO= 8 WRT= 4 MTD= 12	Sensitive = High Level of Risk= High Compliance= High	D1 = HIGH	High
D2= Smart Lenses	F2= Continuous Glucose Monitoring, Drug Delivery, Contact Tracing, Vision Correction	C2= Glucose Levels, Intraocular Pressure, Temperature, Acceleration, Oxygen Levels, Contact Tracing	RPO= 0-1 RTO = 8 WRT = 4 MTD = 12	Sensitive= High Level of Risk= High Compliance= High	D2 = HIGH	High
D3= Pacemakers	F3= implantable battery-powered device which keeps the heart rate from being too slow and helps to treat heart failure	C= Collects data on cardiac events and carries out measurements	RPO = 0-1 RTO = 8 WRT = 4 MTD = 12	Sensitive = High Level of Risk = High Compliance = High	D3 = HIGH	Medium

Table 7: IoT Device Data Protection and Recovery Plan



	Backup Creation				
IT Components	Type of Data Backup	Frequency of Backup	Backup Method	Backup Strategy	Backup Location
D1– Ingestible Sensor	External Receiver, Manual Data Entry, Redundant Sensors, Follow-up testing.	Daily	External Receiver	Full	Sensor with an external receiver
D2– Smart Lenses	Cloud Storage, Local Storage, Synchronization with other devices, Redundant smart lenses	Weekly	Cloud Storage	Incremental	Cloud
D3– Pacemakers	Microprocessors (which can record, process, and transmit information)	Daily	Microprocessors	Full	Information would be stored on the patient records, which are fully backed up on the cloud

6 Conclusion

The primary objective of this research paper is to establish robust security measures within healthcare domains, prioritizing the safeguarding and confidentiality of patient information. In pursuit of this goal, we have developed comprehensive frameworks encompassing risk analysis, a BIA, contingency planning, and detailed tables outlining the identification and mitigation of risks associated with IoT devices, as well as backup procedures. Given the proliferation of IoT devices in the healthcare sector and the continuous evolution of their accompanying software, adhering to the prescribed guidelines and procedures becomes paramount to mitigating security risks and preserving the integrity of user data. Notably, IoT devices often exhibit vulnerabilities owing to their inherent design, necessitating diligent oversight and management by security professionals. Organizations are also advised to conduct a BIA, enabling them to ascertain asset priorities and evaluate the potential impact of security breaches on the organization. By implementing the recommendations delineated in the contingency plan, organizations can enhance their operational efficiency, facilitate data recovery processes, and proactively prevent future security breaches.

References

[1] KR Darshan and KR Anandakumar. "A comprehensive review on usage of Internet of Things (IoT) in healthcare system". In: *2015 International Conference on Emerging Research in Electronics, Computer Science and Technology (ICERECT)*. IEEE. 2015, pp. 132–136.

[2] Yu-Wei Kao et al. "Intelligent search, rescue, and disaster recovery via internet of things". In: *2019 Global IoT Summit (GIoTS)*. IEEE. 2019, pp. 1–7.

[3] Eftychia Lakka et al. "Incident Handling for Healthcare Organizations and Supply-Chains". In: *2022 IEEE Symposium on Computers and Communications (ISCC)*. IEEE. 2022, pp. 1–7.

[4] Frank Maurer. "Agile methods and interaction design: friend or foe?" In: *EICS*. 2009, pp. 209–210.

[5] Suhail Javed Quraishi and Humra Yusuf. "Internet of Things in Healthcare, A Literature Review". In: *2021 International Conference on Technological Advancements and Innovations (ICTAI)*. IEEE. 2021, pp. 198–202.

[6] Trayush Trayush et al. "Iot in healthcare: Challenges, benefits, applications, and opportunities". In: *2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*. IEEE. 2021, pp. 107–111.

OneRuleToFindThem: Efficient Automated Generation of Password Cracking Rules*

Joshua Eckroth[1], Lannie Hough[2], Hala ElAarag[1]
[1]Stetson University, DeLand, FL 32723
{jeckroth,helaarag}@stetson.edu
[2]i2k Connect Inc
dlhough@i2kconnect.com

Abstract

Password cracking tools such as Hashcat support the use of rules that transform a dictionary of words, such as common English words and previously-cracked passwords, into new candidate guesses for hashed passwords. Rules are necessary to achieve high cracking ratios, but they are difficult and time-consuming to build by hand. We have developed an algorithm and implementation that automatically finds successful rules via the combinatorial generation of rules and empirical observation of how often each generated rule transforms a dictionary word to a target password.

Our algorithm includes numerous performance and logical optimizations to avoid the numerous pitfalls that would occur if a naïve brute-force technique were used. In this paper, we explain our algorithm in detail and experimentally compare the performance of its outputs to existing rule sets constructed via various approaches ranging from fully-manual to fully-automated like our own.

We show that our approach achieves comparable cracking performance to other top rule sets while also generating rules that do not exist in other rule sets. This makes cracking attempts using our rules mostly complementary to cracking attempts with other rule sets. We demonstrate that we can achieve top performance by combining our generated rules with other rule sets.

1 Introduction

The most common authentication for applications continues to be passwords. Properly implemented password authentication and very strong passwords, such as long passwords composed of random characters, are generally effective. However, it is well-known that a significant proportion of people elect to use passwords that combine a word with a few numbers or special characters as required by the software. These passwords often have predictable patterns and evolve in predictable ways over time as a user is forced to change their password, often resulting in simple modifications resulting in a new password with a high degree of similarity to the original[2]. While this approach might make passwords more amenable to memorization, it significantly weakens them in the face of a password cracking attack.

A common approach to cracking passwords is through the use of Hashcat[4], utilizing the technique of hashing candidate passwords in a wordlist and checking if the hash matches one found in a list of password hashes. In order to avoid pre-generation of a massive exhaustive wordlist, Hashcat supports the use of 'rules,' which perform transformations on a list of words such as dictionary words and previously-cracked passwords. The resulting transformed passwords are then hashed and the same checks are made. Example rules include reverse (r), append character ($X), and replace (sXY, replace X with Y).

To produce the most guesses and therefore crack the most passwords, one must either have a large wordlist or a large number of rules (or both). However, because most passwords are not generated randomly, some rules will be (sometimes dramatically) more effective than others. Because each additional rule in the list increases the time the cracking process takes to complete, an attacker is incentivized to minimize the number of rules (and wordlist size) while maximizing the percent of hashes that are cracked. The attacker wishes to use only the most effective rules.

Researching how to improve this popular method of cracking is merited on the grounds that it can inform good password policy. Many organizations enforce 'password strength' through length and character diversity requirements, which may provide a false sense of security. If a user meets these requirements simply by appending '123!' to a weak password, that password hash (if it is leaked) might be easily cracked with a common rule. These extra characters do not provide significant protection to a user. If we can find effective password cracking rules, we can create effective password strength requirements by ensuring the rules fail on the user's chosen password.

Many effective rules already exist and some are distributed with the Hashcat software, such as the sizable 'dive' list with about 99k rules. Various approaches have been taken to produce these lists. Some creators manually curate rules, which is a time-consuming process, while others have tried various algorithmic and automated approaches. Often, existing rule sets are aggregated to various degrees to produce a 'super rule,' such as OneRuleToRuleThemAll[11] and also some of the highly-effective 'Pantagrule' lists of rules[10]. The purpose of this paper is to detail a novel fully-automated approach to rule generation, based on an iterative rule accumulation and scoring procedure. We compare the performance of our generated rule sets to existing publicly-available rules. We find that our technique generates new effective rules that others did not, and that the combination of our rule sets achieve the best performance.

The rest of this paper is organized as follows. First, we review related work (Section 2). Next, we describe our algorithms in Section 3. We do this in three parts. First, we show a simple brute-force procedure, then we explain optimizations we made to increase its effectiveness, followed by optimizations we made to increase its performance in terms of time and memory. Then we explain our experimental methodology (Section 4) followed by our results (Section 5) and future work (Section 6).

2 Related Work

Several automated rule-generation techniques are based on the PACK toolkit[7], a collection of tools designed for analyzing password lists to detect masks, rules, character sets, and various other password characteristics that can produce results (rules and masks) designed to work with Hashcat. The 'nsa-rules' analysis by NSAKEY[8] and the rules it generates take advantage of this toolkit, as well as the more effective Pantagrule rules[10].

Pantagrule rule lists were generated using PACK's Levenshtein reverse path algorithm to produce rules which were then sorted by the frequency at which they were generated by PACK. This is similar to the approach NSAKEY took but Pantagrule used a larger set of base passwords to generate rules, which although initially public is now inaccessible. To further optimize the rules, Pantagrule ran the top generated rules against the Pwned Passwords NTLM list[6] using the RockYou wordlist. Ineffective rules were discarded. Several rule lists are created from rules generated by various subsets of the seed data (top passwords, random passwords, and a hybrid of the two).

The 'one.rule' Pantagrule rule list builds upon the OneRuleToRuleThemAll (ORTRTA) rule list[11], which was created by concatenating the top 25% of rules from various other rule lists. 'one.rule' appends top performing Pantagrule hybrid rules to ORTRTA and truncates the list to the size of the 'dive' rules in order to make comparisons against a commonly used rule list of the same size. ORTRTA exceeds the performance of dive on its own, both in total % of passwords cracked and in cracking efficiency on the Lifeboat data dump[11]. Pantagrule's one.rule also compares favorably in total number of passwords cracked against dive at the same total number of rules and against ORTRTA as a superset of its rules, however it is less efficient than ORTRTA as a consequence of containing significantly more total passwords. Pantagrule suggests that their ‘one.rule’ performs better than other known lists the size of dive and they recommend it as a first list to try when cracking hashes.

In addition to traditional rule-based approaches to guessing passwords, some techniques have been developed that attempt to avoid this entirely. PassGAN[5] is an approach that attempts to replace rule-based password guessing with a technique based on deep learning and generative adversarial networks (GANs). A neural network was trained to determine password characteristics and structures without making any assumptions about these. Like our approach, PassGAN makes use of part of the RockYou dataset and trains on it. They then test their results against both RockYou (with training data removed) and a leak of LinkedIn passwords. Their results show that the PassGAN approach is able to match 34.6% of passwords in the RockYou dataset and 34.2% in the LinkedIn dataset. While a typical rule-based attack has the disadvantage of being able to exhaust guesses once all rules have been applied to all initial passwords, PassGAN can generate guesses effectively forever. So while PassGAN can in theory eventually guess more passwords than any other approach, it needs to generate significantly more passwords to do this and can require up to 10x more guesses to reach the same number of matched passwords as competitors. However, PassGAN matches some passwords not matched by any password rule in the rule sets they compared against.

Another approach[9] attempts to leverage representation learning techniques to discover a representation of password distributions. This technique models password representation through a GAN instance and a Wasserstein Auto-Encoder (WAE) instance. Two password guessing frameworks are proposed: conditional password guessing (CPG) and dynamic password guessing (DPG). The model produced by this approach improves on PassGAN against the RockYou test set, cracking 51.8% of passwords in the same number of guesses (5×10^{10}) it took PassGAN to crack 34%. Like PassGAN, the CPG and DPG frameworks guess some passwords that are not cracked by other approaches and DPG allows a guessing attack to focus on unique and otherwise ignored modalities of the target passwords.

Our approach builds rules from the ground up by gradually building complex rules out of simple parts. Rules that are successful are kept, others are eliminated. This approach differs from prior work in the sense that Pantagrule and others generate rules from statistical analysis of passwords, and neural network-based approaches do not generate rules at all (only password candidates).

3 Algorithms

Our rule generator requires two inputs: a set of rule primitives that will be combined to form complex rules, and a set of target passwords such as the RockYou list. We implement an efficient version of what is essentially a brute-force procedure. We first describe the brute-force procedure and then describe our optimizations.

3.1 Brute-force procedure

Given each initial target password (e.g., from RockYou), we apply every primitive rule to the password to generate new passwords. For example, the primitive Hashcat rule ‘r’ (reverse) applied to the initial target password ‘123456’ results in password ‘654321.’ We use a primitive rule set consisting of elementary operations such as reverse (‘r’), remove last character (‘]’), delete all ‘s’ characters (‘@s’), and so on, totaling nearly 400 primitive rules. The selected password is subjected to every primitive, resulting in about 400 new passwords. For each resulting password (such as ‘654321’), we check if it is one of our targets from our initial list of targets (e.g., RockYou). If it is, we boost the score of the rule that was applied. In the end, we have a list of rules with scores indicating which rules were most successful. Initially, we simply boosted the score by 1 for each ‘hit’ but we later adopted an approach that increases the score more for passwords that are harder to hit, discussed in more detail later.

After that initial step of applying rules to a single password, we proceed to choose another password and apply all primitive rules to it, boosting the scores of rules that transform the password to a known target password. A naïve brute-force approach would choose a new password to try either randomly or sequentially from a list of possible candidates but ultimately we choose the next password according to an ordering of all candidates by ‘individual password strength,’ with weaker passwords chosen earlier; details are given below.

Each password that is generated from applying primitive rules becomes a potential candidate itself, unless it is already known from the initial target set. For example, if the rule ‘r’ is applied to ‘foobar,’ producing ‘raboof,’ and ‘raboof’ is not already known from the target set, it becomes a candidate for selection. We record the history of rules that have already been applied, in this case just ‘r.’ When ‘raboof’ is eventually selected as the next password to try, each primitive rule is appended to its rule history, producing complex rules ‘r]’ ‘r @s’ and so on. If ‘]’ applied to ‘raboof,’ which produces ‘raboo,’ is a target, then we boost the score of the complex rule ‘r].’ We note that the initial password ‘foobar’ (pulled from RockYou) was transformed to ‘raboo’ using complex rule ‘r]’ and ‘raboo’ is a target (in this example, though in reality it is not a member of RockYou). Thus, our procedure has discovered a

Algorithm 1 Brute-force procedure, without optimizations

```
$PrimitiveRules \leftarrow$ fileContents(“primitives.rule”)
$Rules \leftarrow PrimitiveRules$
$Targets \leftarrow$ fileContents(“rockyou.txt”)
for all $p \in Targets$ do
    setRuleHistory$(p, \{\})$
end for
$Candidates \leftarrow Targets$
$Processed \leftarrow \{\}$
while $|Candidates| \geq 0$ do
    $p \leftarrow$ chooseOne$(Candidates)$
    $Candidates \leftarrow Candidates \setminus \{p\}$
    $Processed \leftarrow \{p\} \cup Processed$
    for all $r \in PrimitiveRules$ do
        $p' \leftarrow$ applyRule$(p, r)$
        $H \leftarrow \{r\} \cup \{\text{append}(h, r) | h \in \text{ruleHistory}(p)\}$
        setRuleHistory$(p', H)$
        if $p' \in Targets$ then
            for all $h \in H$ do
                if $h \in Rules$ then
                    $s \leftarrow$ getScore$(h)$
                    setScore$(h, s + \text{strength}(p'))$
                else
                    setScore$(h, \text{strength}(p'))$
                    $Rules \leftarrow \{h\} \cup Rules$
                end if
            end for
        end if
        if $p' \notin Processed \cup Candidates$ then
            $Candidates \leftarrow \{p'\} \cup Candidates$
        end if
    end for
end while
```

successful rule that should be utilized in password cracking.

In summary, the brute-force procedure shown in Algorithm 1 begins with an initial list of target passwords and puts them into a candidate set, picks a single candidate password at a time and applies all primitive rules, and boosts the scores of any rules that ultimately produced a password found in the initial list of targets. Each password generated from applying rules goes into the candidate set if it is not already in there, and the sequence of primitive rules that generated it is associated with the password.

Figure 1 shows an example of the combinatorial explosion of candidates that results from the brute-force algorithm. Note that some passwords may be reached by several distinct rule histories, e.g., starting with password ‘123456,’ the password ‘54321’ may be arrived at by applying complex rules ‘r [’ or ‘] r,’ or even ‘$! r [[’ (not shown in the graph).

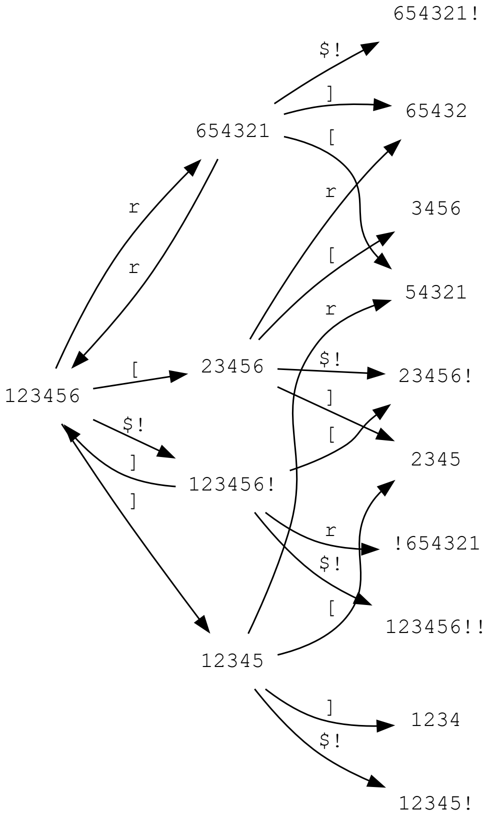


Figure 1: Small example of the combinatorial explosion of passwords generated by applying primitive rules.

3.2 Optimizations for effectiveness

The brute-force procedure suffers from significant drawbacks. Since it lacks any criteria for checking rule validity and structure or for preferring to examine some passwords before others, it is likely to generate worthless rules and take a long time to do so.

3.2.1 Hit a target only once

The RockYou wordlist, which is our input to the algorithm, includes some very basic words like 'password' and even the single letter 'a.' The brute-force procedure will discover rules such as ']]]]]] $a' that will transform any six-character password such as 'gh%@_$' into the password 'a,' and the procedure will boost the score of that rule. But that rule is hardly effective for cracking password hashes. However, the procedure will boost that rule for every six-character candidate because the rule will hit a target (namely, the target 'a'). When we allow this behavior, we see that the procedure yields abundant variations of this logic (erasing characters from either end, then adding a few to hit a small target), and they are not effective in experiments.

An easy way to prevent this behavior is to modify the 'if' block starting on line 17 in Algorithm 1 to what is shown in Algorithm 2.

Algorithm 2 Hit a target only once

```
if $p' \in Targets$ then
    for all $h \in H$ do
        ...
    end for
    $Targets \leftarrow Targets \setminus \{p'\}$
end if
```

3.2.2 Ordering by password strength

Because our algorithm applies all primitive rules to each candidate password, we will produce hits faster if the candidate passwords we choose are those that are the most likely to be transformed into a target password. Intuitively, it makes sense that the application of primitive rules to already very strong passwords, such as long passwords consisting of random characters, would be less effective than the application of rules to weaker passwords. In order to select these weaker passwords earlier in our rule generation procedure we first generate individual password strengths for a sample distribution of passwords.

For individual password strength we utilize a metric invented by Joseph Bonneau called the 'partial guessing metric'[1], which was compared to other metrics and determined to be particularly effective. Important properties of this metric are that it provides equal strength to all passwords in a uniform distribution $\mathcal{U}_N$ where each of N events are equally likely and that it rates any event more weakly than events less common in the population-wide distribution χ.

This metric is developed with the assumption that the population-wide distribution χ of passwords is completely known and addresses the issue of estimating the strength of previously unseen passwords when a sample is used as an approximation of χ. For our approximation we use as a sample the passwords in the RockYou dataset and the frequency at which they appear. We produce a mapping of the passwords in our distribution to their strengths and provide for estimating the strength of unseen passwords, allowing us to assign each candidate encountered with a strength value.

With strength values known for all initial candidates and the ability to determine the strength of new candidates, we can create a priority queue where high priority candidates are those with a low strength. We select these weaker candidates first, resulting in more hits of target passwords. We previously mentioned that rules have their scores increased when we hit a target password. Instead of simply incrementing the score for all rules equally, we made the decision to use the password strength of the hit password as the number by which the rule score is incremented. This yielded slightly better results than incrementing rules equally on every hit. Intuitively this makes sense. Weaker passwords are likely to be reached by considerably more rule variations and a very long or complicated rule that hits a small number of weak passwords is likely of little utility; the passwords probably will be reached by other rules as well. A rule that gets hits on very strong passwords on the other hand is likely to significantly increase coverage, so it makes sense to give these rules a boost.

3.2.3 Rule simplification

The brute-force procedure appends each primitive rule to each rule in a password's rule history on line 15 of Algorithm 1. For example, if the password 'password123' was reached by iterative appending of primitive rules '\$1,' then '\$2,' then '\$3,' the password will have rule history '\$1 \$2 \$3' (among others, possibly). If 'password123' is later selected as a candidate, each primitive rule will be added to the end of that history and tested to see if it hits a target. For example, '\$4' will be added and since 'password1234' is a target, each rule in the history (with '\$4' appended) will be boosted. Thus, the rule '\$1 \$2 \$3 \$4' will be boosted.

We have identified numerous conditions in which complex rules (sequences of primitive rules) are equivalent to a simpler rule. For example, the rule '\$1] \$a' is equivalent to '\$a.' We also normalize rules by reordering some sequences of primitives. For example, the rule ‘^2] ^1’ is equivalent to ‘^2 ^1]’ (they both insert ‘12’ at the front and remove the last character). If we normalize all rules according to some common simplification and sequencing logic, we can be sure to boost the normalized version of a rule instead of boosting different variations and thus lowering the score of the rule.

We have about 50 rule simplifications that are specified as regular expressions. Table 1 shows the number of rules generated originally (without simplification), and the number after simplifying, for different lengths of rules. Rule length indicates number of primitives in each complex rule; e.g., ‘r] $1’ has length 3. Original count specifies the number of rules generated with a certain length, without rule simplification. Simplified count shows number of rules that remain after rewriting some to a simpler form. Simpler forms will typically be repeats and will be removed from the count. It is clear that exponential growth is still present as the rule length increases. However, we benefit by ensuring we are scoring the normalized rule rather than equivalent variations.

Rule length	Original count	Simplified count	Ratio
1	313	313	1.0
2	97,000	90538	0.93
3	30,762,000	26,726,754	0.87
4	296,735,000	255,805,952	0.86

Table 1: Impact of rule simplification.

We modify the brute-force procedure with this optimization at line 15 by first simplifying the new complex rule before adding it to the rule history. This change is shown in Algorithm 3.

Algorithm 3 Rule simplification

```
$H \leftarrow \{r\} \cup \{\text{simplify}(\text{append}(h, r)) | h \in \text{ruleHistory}(p)\}$
```

3.2.4 No-op rule detection

We also detect rules that accomplish nothing. For example, the rule ‘r r’ (reverse, then reverse again) will be boosted repeatedly since it essentially does not transform a password at all. If the candidate password is already a target, then the password generated by this rule is also a target (because it is the same word), so ‘r r’ will be boosted. In effect, the procedure will yield abundant high-scoring rules that accomplish very little and will not be effective for cracking hashes. These no-op rules are detected and eliminated as shown in Algorithm 4

Algorithm 4 Eliminate no-op rules

```
$H \leftarrow H \setminus \{h | h \in H : \text{isNoOpRule}(h)\}$
```

3.2.5 Inventing primitive rules

In order to facilitate generation of complex rules, we promote a complex rule to the primitive rule set if the rule produces a target sufficiently often (we experimentally chose this threshold to be 10 targets). For example, if the primitive rule '$3' is added to a rule history containing '$1 $2' and the resulting complex rule '$1 $2 $3' produces a target at least 10 times, then '$1 $2 $3' is added as a primitive. Hence, it will be added as a single unit to other rules, e.g., it will be added to '$1 $2' as in this example, yielding '$1 $2 $1 $2 $3.' In our experimental results (Section 5), we will show how many new primitive rules are invented.

3.3 Optimizations for memory and time

In this section, we explain how we ensure our algorithm uses limited memory and reduces runtime.

3.3.1 Use of radix trees

Because a password (potential new candidate) can be reached by many combinations of primitive rules applied to a candidate, it is important for our procedure to recognize which of these potential new candidates have already been processed in order to avoid significant duplicate processing. The naïve approach of using a set very quickly becomes untenable with rapid growth in memory consumption. To mitigate this, our procedure takes advantage of radix trees to store unprocessed and processed passwords. The substring 'password' in 'password123', 'password!1', and 'passwordxyz' will only be stored once. This optimization dramatically slows down memory consumption as our process proceeds.

We make use of the same optimization to store our large number of generated rules.

3.3.2 Capping the candidate set

The main growth of memory in the brute-force procedure is the result of generating new password candidates. These candidates are saved to the queue and processed according to the main loop starting on line 9 of Algorithm 1. In practice, we specify a maximum number of cycles (i.e., how many times to repeat that loop), and we also choose a batch of candidates at a time. We typically run for 1,000 cycles and choose 400 candidates at a time. We can compute the number of password candidates that will ever be examined as the product of these two numbers (400,000). Whenever a password is generated and it was not previously known, it is scored according to its strength and added to a priority queue. Scores do not change after candidates are added to the queue, so periodically (say, every 100 cycles), we eliminate any members of the queue that are below the 400,000th position.

This technique allows us to cap the size of the candidate set. Doing so causes the memory requirements to grow in terms of the size of the radix tree storing the list of generated rules instead of the size of the candidate set. While the brute-force procedure suffers from excessive growth of memory due to the combinatorial nature of password generation, this efficient variant reduces the resident memory size, thus allowing a significantly longer run time, which results in not only more rules but a better ordering of rules based on their scores.

4 Experimental Methodology

We chose to use the full RockYou wordlist containing about 14 million plaintext passwords. This is our target set during training (rule generation), and the original set of candidates in our procedure. We use a different, much more extensive, set of hashed passwords to evaluate the performance of our approach.

In order to utilize our password strength metric, we require a target list that is sorted by frequency of occurrence in real-world usage. RockYou is not sorted in this way. The Pwned Hashes list[6] includes frequencies, but not plaintext passwords, so we hashed each RockYou password and looked up its frequency in the Pwned Hashes list, and ordered RockYou by those frequencies.

We ran the algorithm for 1,000 cycles and 400 candidates per cycle, resulting in 400,000 passwords being analyzed. Recall that for each analyzed password, all primitive rules are applied, generating far more candidates than can ever be analyzed.

Our algorithm produces a list of rules. We remove logical duplicates using the 'duprule' program[3], which catches some duplicate rules that our rule simplifier misses. For example, it finds that 'r] ^n' (reverse, remove last, add 'n' to front) is the same as '$n r]' (add 'n' to end, reverse, remove last), so the latter rule is removed. With these deduplicated rules, we use Hashcat and the same RockYou wordlist to attempt to crack the most frequent 100 million Pwned Hashes[6]. We record the percent cracked.

We compared performance of our rules, named *OneRuleToFindThem* (ORTFT), against several other lists of rules, including some that incorporate our own rules:

- An empty rule list, to see what percentage RockYou itself can crack.
- Rules generated from the PACK algorithm[7] on all of RockYou input, then trimmed to various sizes.
- Different sizes of our generated rules, ordered by rule score: top 64 rules, top 10,000 rules, top 100,000.
- The 'best64' and 'dive' rules that come with the Hashcat distribution.
- OneRuleToRuleThemAll (ORTRTA)[11], a combination of other rules: d3adhob0.rule, hob064.rule, KoreLogicRulesPrependRockYou50000, _NSAKEY.v2.dive.rule, and generated2.rule by oclHashcat v1.20.
- OneRuleToRuleThemAll with our generated rules appended, then trimmed to the size of the 'dive' ruleset (99k rules) for comparison purposes, which we refer to as *OneRuleToFindThem* (ORTFT).
- Pantagrule's[10] top-performing rule list, pantagrule.private.v5.popular. Pantagrule rules are generated by running PACK on a private massive set of plaintext passwords, or alternatively, a similarly large set from hashes.org, which is no longer available. Since we do not have access to these sets, we trained PACK on RockYou and call this list of rules PACK, mentioned above.
- Pantagrule's rules pantagrule.private.v5.popular plus our generated rules, with duplicates removed from the combined set.

Note that we try various sizes of our generated rules and PACK generated rules, because we know that these rule are ordered by some kind of score. Other rules from the community are not guaranteed to be ordered.

We also check the number of duplicate rules (according to the 'duprule' program[3]) that we share with other rules like OneRuleToRuleThemAll, dive, and Pantagrule-popular. If we find that our generated rules are mostly duplicates of other rules, then our technique is not adding much diversity and therefore not much value. If, however, we find we do not share many rules in common, then our procedure is finding new rules that may be used to complement existing rules created by others.

Generally speaking, one can crack more password hashes by trying more guesses. More rules with the same word dictionary produce more guesses, so more rules generally result in more cracked passwords. However, there are diminishing returns. One rule set might be able to earn a certain percent cracked hashes while another rule set might be able to achieve the same percent but with fewer rules. The latter rule set will run faster since there are fewer guesses.

We borrow the rules-per-percent metric (RPP) from Pantagrule[10], which is a measure of the average number of rules required to crack a single percent of the hashes for a given set of hashes. We subtract the percent of cracked hashes one obtains by using no rules, i.e., using the RockYou word dictionary on its own (which cracks 6.33% of the top 100 million Pwned Hashes). RPP is defined as,

$$RPP = \left\lfloor \left(\frac{\|Rules\|}{100 * \frac{\|Cracked\|}{\|Hashes\|} - 6.33} \right) \right\rceil$$

where $\lfloor \rceil$ rounds to the nearest integer.

5 Results

A short list of the highest-scoring rules generated by our algorithm is shown in Table 2. These rules match our intuition about how people typically modify an old password or dictionary word to make a new one. We also did a run of Hashcat in 'debug' mode with our top 50k rules, which allows us to record which specific rules in our rule set were responsible for cracking the most passwords. Included in the top ten most successful rules were nine of our highest scoring rules (all except 'remove last character').

As shown in Table 3, while it is not the case that selecting only a 'top-n' set of rules from our generated rules produces a clear win against some common similarly-sized rule sets like 'dive' (99k) or various Pantagrule rule sets, our results clearly indicate that we are generating some strong results with many rules that are not included in these existing lists. Interestingly and somewhat surprisingly, given the relative ease of hand-curating small rule lists, we do exceed best64 in performance with our top 64 generated rules.

In rarecoil's analysis of their Pantagrule rule sets they compare 'dive.rule' to a combination of OneRule and generated Pantagrule rules ('one.rule', equal in size to dive) in order to showcase the utility of their rules. At the time of its creation Pantagrule's 'one.rule' performed better than known lists equal in size to dive and as far as we know this is still true today.

Our results *OneRuleToFindThem* compared to dive and Pantagrule's highly effective 'one.rule' of the same size demonstrates that we are more effective than dive, cracking around 5% more passwords, and almost as effective as 'one.rule,' cracking only 2% fewer passwords. In nearly reaching the performance of the most effective known list of this size, we show that our approach is effective. Despite being marginally less effective as a single rule set, our approach uses many rules that Pantagrule does not; this makes our approach to generating rules complementary to Pantagrule's and using both rule files during a cracking attempt should yield good results. Pantagrule notes that their approach has

Rule	Score	Explanation
$1	508,091	Add ‘1’ to end
T0	369,973	Toggle case of first character
$2	355,021	Add ‘2’ to end
t	313,526	Toggle case of all characters
$7	290,926	Add ‘7’ to end
$3	284,959	Add ‘3’ to end
]	281,415	Remove last character
$1 $2	273,308	Add ‘12’ to end
$5	253,183	Add ‘5’ to end
$4	246,386	Add ‘4’ to end

Table 2: Top rules generated by our procedure. Scores represent relative success at matching target passwords (an approximation of cracking success).

only a couple-thousand rule overlap with OneRuleToRuleThemAll; this is true of our *OneRuleToFindThem* as well.

While this comparison highlights the effectiveness of our *rules* in comparison to the Pantagrule rule set, it do not necessarily indicate the effectiveness of our *procedure* compared to the procedure used to generate the Pantagrule rule sets. This is because while we used RockYou as our wordlist, Pantagrule was developed with the use of a public but now inaccessible wordlist (containing about 57 times as many words as RockYou). However, as Pantagrule describes their procedure we can repeat their rule generation using the same wordlist we used, producing effectively our own version of Pantagrule's rule files. These are labeled in Table 3 as the ‘PACK top-n’ rules. If one compares our top-100k rules to the PACK top-100k rules, we are within 1%, indicating that our procedure is more effective than the comparison of our rules only to Pantagrule's final rules might suggest. Furthermore, we once again excel in the class of very small rule sets, besting ‘PACK top-64’ with our top 64 rules.

Overall, we have demonstrated both the efficacy of our own approach to generating rules and produced a set of rules that would be beneficial to use in tandem with rules produced from other top rule lists.

In Figure 2, we see some select rules plotted according to the number of attempted guesses vs. cracked hashes. As Hashcat runs, each word in the wordlist (RockYou) is given to each rule, generating some number of guesses (equal to the number of rules). Then the next word in the wordlist is tried in the same manner. We run Hashcat on a version of RockYou that is ordered according to frequency as given by the Pwned Hashes dataset[6]. We see in the plot that small rule sets like our top-64 rules have a nice trend in the early number of attempts. This is a result of running through RockYou more quickly with just 64 variations of each RockYou word being tried. However, of course, with only 64 rules, only 25% of the hashes are cracked. Switching to the large Pantagrule-popular + Ours rule list, we see it takes more time to crack an equivalent number of hashes as the 64-rule list, but ultimately finds far more. Curiously, OneRuleToRuleThemAll (ORTRTA) also finds a good amount but has poor performance early on. We have not studied the makeup or organization of ORTRTA enough to understand this behavior.

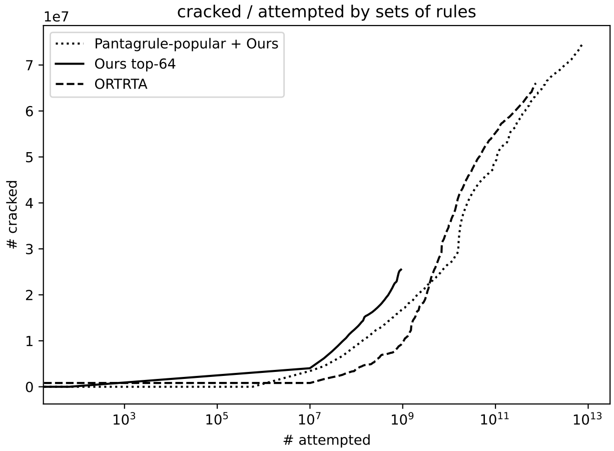


Figure 2: Number of cracked hashes per attempted guess, for select rule sets. Note the x-axis is logarithmic.

As a means of determining if a set of rules is efficient at cracking, we use the rules-per-percent metric (RPP). This metric penalizes larger rule sets that crack the same number of hashes as smaller ones. Table 3 shows the size, cracked %, and RPP for various rules, and Figure 3 plots the relationship between number cracked and RPP. In the figure, the best place to be is near the top left: more cracked, lower RPP.

Examining Table 3, we can summarize these findings by picking the best rule for achieving a certain cracked % while minimizing time (so minimizing RPP). Table 4 shows these results.

Figure 4 shows that the number of complex rules grows per cycle, but gradually levels off. Recall that a complex rule is created when it has never been seen before and is able to transform a candidate password into a target. Over time, fewer rules are generated that are both novel and successful. Also recall that particularly successful rules are promoted to primitives. The frequency of this occurrence also levels off, as shown in the figure.

Rules	Count	Cracked	RPP
No rules (RockYou itself)	0	6.33%	N/A
PACK top-64	64	24.57%	4
best64	64	24.99%	5
Ours top-64	64	25.49%	3
Ours top-10k	10,000	53.76%	211
PACK top-50k	50,000	61.13%	912
ORTRTA	51,998	66.03%	871
dive	99,092	64.71%	1697
ORTFT (ORTRTA + Ours)	99,092	69.92%	1558
Pantagrule-one-royce	99,092	71.93%	1511
Ours top-100k	100,000	62.79%	1771
PACK top-100k	100,000	63.92%	1736
Pantagrule-popular	478,736	73.98%	7077
Pantagrule-popular + Ours	574,487	74.84%	8385

Table 3: Rules-per-password cracked metric (RPP) for various rule lists, ordered by size of the list and then by cracked %.

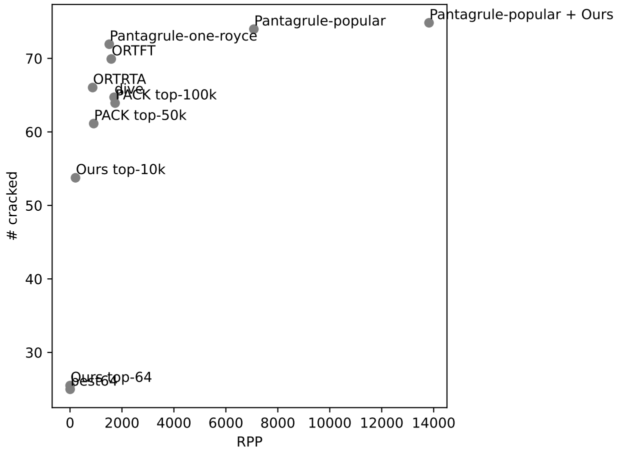


Figure 3: Number of cracked hashes per RPP value, for various rules.

The 'duprule' program [3] eliminated 34,365 duplicate rules from our generated set of 529,536 rules (6.5%). Table 5 shows how many deduplicated rules generated by our procedure are also found in various other rule lists. 'ORTRTA' represents the rule set 'OneRuleToRuleThemAll' [11]. 'Pantagrule

% cracked	Best rules	RPP
> 5%	No rules (RockYou itself)	N/A
> 20%	Ours top-64	3
> 50%	Ours top-10k	211
> 60%	PACK top-50k	912
> 65%	ORTFT (ORTRTA + Ours)	1588
> 70%	Pantagrule-one-royce	1511
Max	Pantagrule popular + Ours	8385

Table 4: Summary of cracked % and RPP values for top rules.

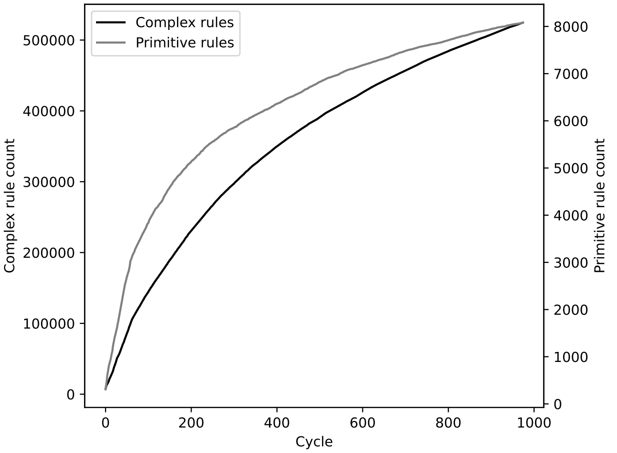


Figure 4: Growth of complex and primitve rules over time (cycles). As targets are hit, more complex rules are added.

popular' refers to Pantagrule's 'pantagrule.private.v5.popular.rule' [10]. The 'Count' column indicates the count of rules in the rule set, the 'Duplicates' column indicates the count of rules in the rule set that match one of our generated rules, and the 'Pct. dup.' column is defined as the 'Duplicates' column divided by the 'Count' column.

The low 'percent dup' values indicate that our generated rule set does not have much overlap with existing large rule sets. Thus, our techniques compliment each other, and the best cracking performance is obtained by combining rule sets.

In the early cycles of the algorithm, common passwords are selected from the candidate set, and primitive rules are applied to them to generate new

Rules	Count	Duplicates	Pct. dup.
best64	64	47	73.4%
ORTRTA	51,998	5,318	10.2%
dive	99,092	3,712	3.7%
PACK-100k	100,000	4,919	4.9%
Pantagrule popular	478,736	11,227	2.3%

Table 5: Counts of rules that are found in both our generated rules and each existing rule set.

passwords. We check if each generated password matches a target password, and if so we call it a 'hit.' Once a target password is hit, it is no longer considered a target. Since many passwords in the RockYou list are simple variations of other passwords in the list, we hit a lot of targets early but fewer over time. This trend is shown Figure 5. The decline in hit percent appears to be exponential.

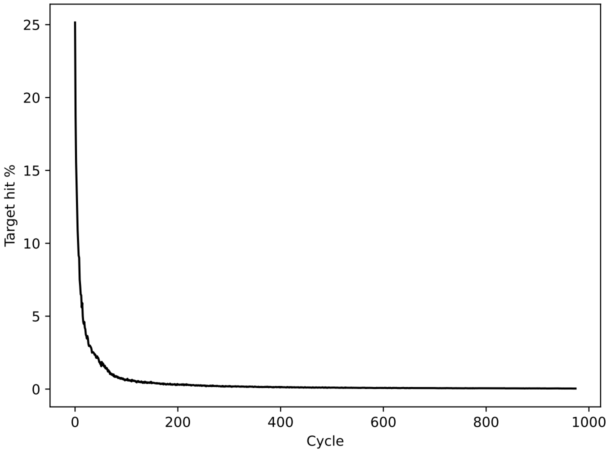


Figure 5: Percent of hits per cycle.

Figure 6 shows the time required per cycle. As the number of cycles increases, more rules have been generated and stored in the radix tree, thus requiring more work to find and add rules. The spikes are due to the time required every 100 cycles to reduce the queue of candidates (a priority queue) to limit memory growth.

Figure 7 shows the growth of resident memory over time. The growth is

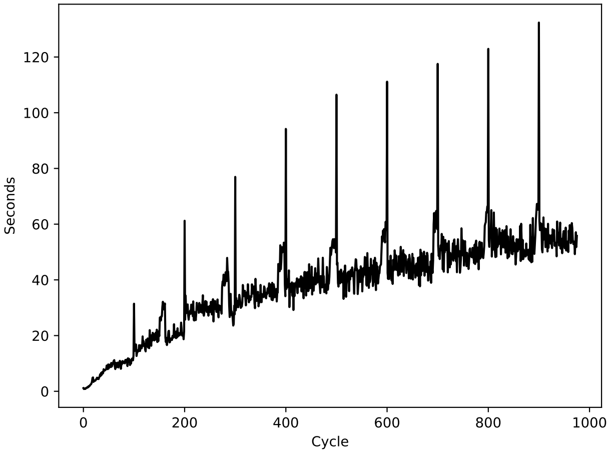


Figure 6: Time per cycle.

logarithmic due to the capped candidate set size and the logarithmic growth of the radix tree storing generated rules. The memory plot is not smooth because each 100 cycles memory is reduced by trimming the candidate set size. However, the process keeps this free memory space reserved for some time rather than releasing it back to the operating system.

Because the growth is logarithmic, we avoid the excessive memory use required by a simple brute-force procedure. Note, however, that the memory usage grows past 60 GB, which is significant for consumer-grade computers. Memory usage can be reduced by running the algorithm for fewer cycles (specified by a 'max cycles' parameter) and/or fewer password candidates chosen per cycle (also specified by a parameter).

When we consider Figure 4 (rule growth per cycle), Figure 5 (hit percent per cycle), Figure 6 (seconds per cycle), and Figure 7 (memory per cycle) all together, we see that our algorithm expends a growing amount of resources to generate a decreasing number of rules. The algorithm produces diminishing returns. This is to be expected: the 'easy' and most successful rules are found early, while uncommon rules that hit password targets that are infrequent (passwords that rarely appear in the Pwned Hash set) are found rarely and only after extensive searching.

The same phenomenon can be observed in Table 3, which shows the 'rules-per-password cracked' (RPP) metric for various rule lists. This metric estimates the number of rules required to crack a single password. With very small lists of just the most effective rules, such as our top-64 or best64 from

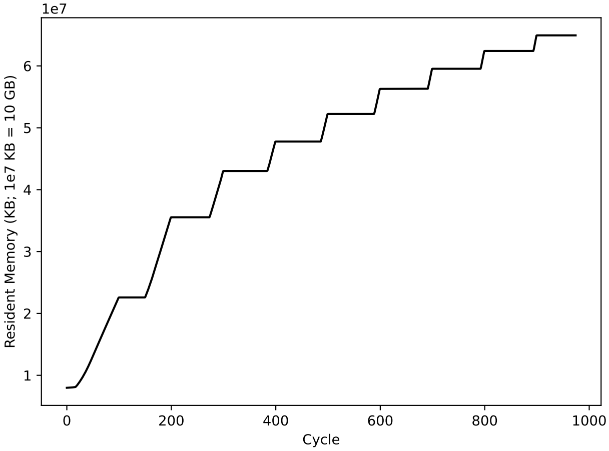


Figure 7: Resident memory per cycle.

Hashcat, one can crack a significant portion of hashes with minimal effort. These are the ‘easy’ hashes. The long-tail of rare passwords are much harder to crack and require more work the greater their rarity.

6 Future Work

In order to get a better picture of RPP trends, one of the more important metrics in analyzing the efficacy of rule sets, more research and experimentation with different data sets and more sizes would be beneficial.

Interestingly, our research has also revealed that many existing lists of rules include many rules that are functionally duplicates of each other, making the cracking process unnecessarily longer. ‘duprule’ or a similar program should probably be run against most existing rule lists before using them in a cracking attempt, and more research into making sure rule lists contain as few functional duplicates as possible is certainly warranted. More research is merited in the area of preventing the generation of duplicate Hashcat rules with existing automated approaches.

7 Conclusion

Our approach and results are interesting for several reasons. Our algorithm follows a bottom-up rule generation procedure which is able to discover rules that other approaches such as reverse-engineering passwords to find rules or neural network-based approaches fail to find. The rules generated by our algorithm have an immediate practical use. They are best combined with other rulesets so that the greatest number of passwords may be cracked since they have little overlap with other rulesets. Since the overlap is small, we suspect that there are still a large number of unexplored rules that would be highly effective, warranting further research in the areas of rule generation and ranking both with our approach, and with other approaches.

Acknowledgments

We wish to thank Jonathan Lamoureux for his contributions to this project.

Availability

Our code and results are available on GitHub at `github.com/joshuaeckroth/passwords`.
We used various datasets to generate our results:

- RockYou plaintext passwords: `github.com/zacheller/rockyou`
- Pwned Passwords version 8, ordered by prevalence: `haveibeenpwned.com/Passwords`
- Pantagrule rules: `github.com/rarecoil/pantagrule`
- Common English words: `github.com/alex-pro-dev/english-words-by-frequency`

Hashcat was used to measure the performance of rules: `github.com/hashcat/hashcat`. We also used ‘duprule,’ a duplicate rule detector: `github.com/mhasbini/duprule`.

References

[1] Joseph Bonneau. "Statistical metrics for individual password strength". In: *Security Protocols XX: 20th International Workshop, Cambridge, UK, April 12-13, 2012, Revised Selected Papers 20*. Springer. 2012, pp. 76–86.

[2] Ameya Hanamsagar et al. "Leveraging semantic transformation to investigate password habits and their causes". In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 2018, pp. 1–12.

[3] M. Hasbini. *duprule: Remove duplicate Hashcat rules*. `https://github.com/mhasbini/duprule`. Accessed: 2023-02-05.

[4] Hashcat. *Hashcat*. `https://hashcat.net/hashcat/`. Accessed: 2023-02-07.

[5] Briland Hitaj et al. "PassGAN: A deep learning approach for password guessing". In: *Applied Cryptography and Network Security: 17th International Conference, ACNS 2019, Bogota, Colombia, June 5–7, 2019, Proceedings 17*. Springer. 2019, pp. 217–237.

[6] Troy Hunt. *Pwned Passwords*. `https://haveibeenpwned.com/Passwords`. Accessed: 2023-02-05.

[7] Peter Kacherginsky. *PACK*. `https://github.com/iphelix/pack`. Accessed: 2023-02-07.

[8] NSAKEY. *nsa-rules*. `https://github.com/NSAKEY/nsa-rules`. Accessed: 2023-02-07.

[9] Dario Pasquini et al. "Improving password guessing via representation learning". In: *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2021, pp. 1382–1399.

[10] rarecoil. *Pantagrule: Gargantuan hashcat rulesets generated from compromised passwords*. `https://github.com/rarecoil/pantagrule`. Accessed: 2023-02-05.

[11] Not So Secure. *One Rule to Rule Them All*. `https://notsosecure.com/one-rule-to-rule-them-all`. Accessed: 2023-02-05.

Data Protection and Recovery Plan for Retail IoT Domain[*]

Syed Rizvi, Justin Moore, Zachary McKee, Mark Ihnat
Department of Information Sciences and Technology
Pennsylvania State University, Altoona, PA 16601
{srizvi, jtm6141, zvm5146, mzi70}@psu.edu

Abstract

Information security holds significant importance in numerous large organizations. However, the retail sector, particularly small organizations, often lacks the necessary reinforcement in terms of information security, risking serious implications on customer financial information, business owners, and supply chain management. This research addresses this critical issue by focusing on the implementation and application of a risk management (RM) framework and a business impact analysis (BIA) within the retail domain, with a specific focus on a Dollar General store. By leveraging the RM framework and BIA model, this study comprehensively examines the identification, analysis, and prioritization of critical business processes, their recovery criticality, resource requirements, and recovery priorities. The findings will be documented, and recommendations will be provided for optimal implementation of the BIA model within an IoT network for Dollar General stores or any retail organization. These insights will enable the formulation of a comprehensive data protection and recovery plan. This research underscores the urgent need for small retail organizations to enhance information security measures to safeguard customer financial information, protect business owners' interests, and ensure robust supply chain management. Implementing the RM framework and BIA model will assist in mitigating risks and establishing resilient information security practices in the retail IoT landscape.

1 Introduction

The retail industry has witnessed numerous transformative changes over the years, revolutionizing the shopping experience. This holds true for grocery stores, including Dollar General, as highlighted in the present paper. Earlier advancements encompassed the introduction of the Universal Product Code (UPC), commonly referred to as the barcode, which significantly facilitated inventory management [6]. More recently, the integration of Internet of Things (IoT) devices has had a profound impact on the retail sector, benefiting both customers and employees alike. Notably, inventory management software equipped with IoT capabilities enables real-time monitoring of inventory levels, providing accurate and up-to-date information on stock availability. The adoption of RFID tags for inventory tracking has proven to be an effective practice, empowering managers to ascertain the need for restocking and reorder items accordingly. This approach also contributes to enhancing supply chain management by tracking shipments and improving overall visibility, which is particularly crucial given the challenges experienced in the supply chain domain in recent years. The implementation of RFID systems further augments product visibility and traceability, aligning with supply chain and inventory management software [3]. In conjunction with these advancements, some companies have employed robots to perform tasks such as shelf stocking and inventory management [6]. Asset Protection departments also benefit from IoT devices, including surveillance cameras integrated with facial recognition technology, which aid in preventing shoplifting incidents and reducing product and revenue loss. Numerous popular retail stores have also developed dedicated mobile applications that leverage location-based services. Through these applications, customers can effortlessly search for product locations, check stock availability, and view pricing information.

Moreover, customers now have access to novel shopping methods. Self-checkout systems have witnessed a recent surge in popularity, with certain stores transitioning to fully automated self-checkout processes, eliminating the need for cashiers. This not only reduces the time spent during the checkout process but also enhances overall operational efficiency. In addition, an alternative form of grocery shopping has emerged in the form of curbside pickup, catering to customers who prefer not to browse through the store physically. This method allows customers to place their orders online and schedule a designated pickup time. Upon arrival, the customer parks their vehicle, and store employees bring the ordered items directly to their car. This service encompasses a wide range of products available at the store. Both of these options significantly reduce waiting times and contribute to improved operational efficiency as a whole. Research indicates that approximately 28% of retailers have already integrated Artificial Intelligence/Machine Learning (AI/ML) technologies into their operations. Additionally, a survey suggests that 49% of respondents anticipate cost reductions within the supply chain, while 44% expect increased productivity as a result of implementing AI solutions [1, 5].

The retail domain continues to face vulnerabilities and threats, encompassing both digital and physical challenges. One notable concern is the prevalence of weak passwords in retail systems, often remaining unchanged from their default state [4]. Even when change is possible, the complexity of passwords is typically limited. Additionally, devices within retail stores are frequently left unpatched, creating open vulnerabilities that can be exploited by attackers. Exploitation of these vulnerabilities can lead to the unauthorized collection of sensitive customer data, including banking information, addresses, phone numbers, and names. Denial-of-Service (DoS) attacks, both in digital and physical forms, are also prevalent risks. Inadequate encryption algorithms, open ports, and vulnerable services can provide avenues for such attacks [4]. By seizing control of devices, attackers can overload target systems, causing them to slow down or cease functioning entirely. Physical attacks manifest in various ways, such as employees stealing equipment from another department, hampering efficient operations.

Additionally, physical disruptions like power outages resulting from storms can incapacitate a store, affecting customer checkout and order pickup processes. Such outages also impact the functionality of freezers and coolers, posing risks to perishable food items. Attackers may also install skimmers, surreptitiously capturing credit or debit card information without the cardholder's knowledge. Insider threats pose a significant concern, as some employees may engage in theft of food items, resulting in revenue loss and misleading inventory management software in the long run. Unauthorized access to data and misuse of company time are other potential risks certain employees pose. To mitigate these risks, companies must implement measures to prevent unauthorized access to customer data and provide comprehensive security training to employees [1]. Given the multitude of vulnerabilities and threats present in the retail IoT domain, it is crucial to maintain up-to-date devices to ensure their security.

Considering the array of vulnerabilities and threats discussed, it is imperative to develop a comprehensive contingency plan and data recovery plan. These measures ensure the secure storage of data and establish backup protocols to minimize data loss. Given the ever-expanding presence of IoT devices in the retail domain, a robust data recovery plan becomes indispensable in mitigating the risks associated with the aforementioned threats. Failure to address these risks adequately could expose a company to potential liabilities, legal repercussions, and reputational damage in the event of a breach. By meticulously formulating and implementing detailed plans, an organization demonstrates its unwavering commitment to safeguarding the well-being of stakeholders and consumers as its highest priority. Notably, an article highlights that the reported instances of cybersecurity incidents with potential ramifications for government, private sector, and personal information exhibited a staggering 782% increase from 2006 to 2012 [2]. The frequency of such incidents has only continued to rise in subsequent years.

This paper is composed with the primary objective of emphasizing the significance of IoT devices in the retail domain and their associated security considerations. The subsequent sections of this paper incorporate several tables and figures to illustrate a network map of Dollar General and outline corresponding recovery plans. The initial phase of this study involves conducting a case study on Dollar General, wherein a risk management framework is established. Subsequently, a comprehensive BIA outlines the sequential steps undertaken within the BIA model. Finally, a robust data protection and recovery plan is formulated to guarantee the utmost security of the data.

2 Case Study for Retail IoT Domain

This case study presents a hypothetical analysis of the risk management framework of a Dollar General store as an example. The objective is to identify vulnerabilities within the framework and provide recommendations to address these issues. By examining the risk management practices in this hypothetical scenario, valuable insights can be gained for enhancing information security in the retail sector. The study encompasses the application of a risk management (RM) framework and a business impact analysis (BIA) model, specifically tailored to the retail domain. During the assessment of the Dollar General store, several security concerns regarding IoT devices were observed. The company's policies pertaining to the protection of its IoT devices are deemed weak. Of significant concern is the absence of encryption on all network-connected devices, including card readers, which are crucial for secure credit card transactions. This poses a major risk as unauthorized individuals could potentially access these IoT devices and covertly collect credit card information from unsuspecting customers. Such a breach could result in substantial financial losses and make the company liable for significant instances of identity theft. Additionally, the reputation of the company could be severely impacted depending on the duration of this undetected security vulnerability. This represents just one aspect of the issues identified within the framework of Dollar General stores. Another consideration is whether customers should be granted access to the public Wi-Fi provided by the company. Figure 1 depicts a network diagram illustrating the infrastructure of a Dollar General store.

The provision of network access to individuals introduces numerous security

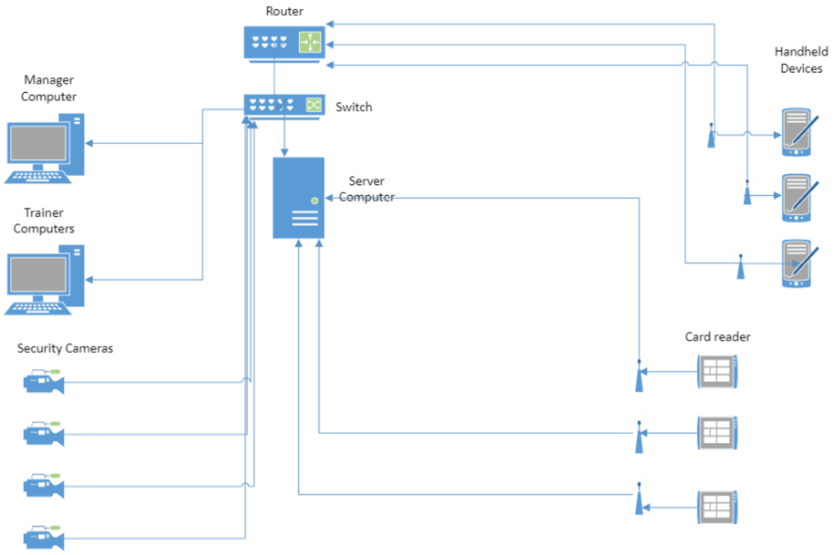


Figure 1: Dollar General Store Network Diagram

vulnerabilities. Granting people access to the internet provides them with an opportunity to conduct system scans and reconnaissance, enabling them to plan potential attacks. Upon entering the store, it is evident that the room housing the servers, routers, and switches is accessible to anyone at any time. Since human actors pose the most significant threat to any system, this raises concerns about the potential existence of backdoors left by individuals who have accessed the server. Implementing physical security measures represents one of the fundamental aspects of enhancing system security. Notably, none of the cameras installed around the store employ encryption. This poses a security risk as it enables potential attackers to execute man-in-the-middle attacks, allowing them to intercept and manipulate video footage. For instance, an attacker could delete video evidence of a theft, resulting in a complete loss of revenue and impending legal action against the perpetrator. These identified risks are merely a subset of the vulnerabilities observed within the particular Dollar General store under examination. To comprehensively assess these risks, we will undertake a BIA to determine the potential impact of a security breach on the business. This analysis will enable us to identify the critical business functions requiring contingency planning.

3 Risk Management Framework

Our risk management framework encompasses five fundamental steps, as illustrated in Figure 2 below. These steps serve as a structured approach to conduct a comprehensive risk assessment specifically tailored for a Dollar General store.

3.1 Risk Identification

The initial step in risk identification involves identifying the components of the information system. Within a Dollar General store, these components encompass various elements, including employees (register clerks/stockers), a shift manager, item and sales data, cash register software, an inventory management system, a handheld scanner, a credit card reader, a cash register, a database server, wireless access points, a network switch, and a router. Each of these components represents an asset with differing levels of criticality. These assets can be categorized into two main groups: people and information systems. Among the assets, there are vulnerabilities associated with credit card readers, data servers, wireless access points, network switches, routers, and, notably, individuals. Several threats are prevalent in relation to these vulnerabilities, such as skimming devices, SQL injections, inadequate encryption, data breaches, piggybacking and evil twin attacks, DHCP spoofing and starvation attacks, weak passwords, and social engineering tactics.

3.2 Risk Evaluation

This phase entails assessing the assets and threats present within the organization. In the context of Dollar General, our evaluation has identified vulnerabilities in credit card readers, database servers, wireless access points, network switches, routers, and personnel. To determine the asset value of these components, we will utilize their average cost, which will be presented in Table 1. Subsequently, it is essential to document the evaluation of threats and their corresponding probabilities of occurrence, as outlined in Table 2. Regarding the threats faced by Dollar General, the probabilities associated with each threat are as follows: Then the calculation of the Risk Magnitude for each asset must be done in order to prioritize the risks properly. Table 3 presents the Risk Magnitude for each asset, derived by multiplying the probability of a risk event occurring (expressed as a percentage) by the corresponding asset value. The asset values can be found in the preceding table, while the risk probability percentages are provided in the current table. This calculation assumes significance as it enables the prioritization of risks, ensuring that the most critical ones receive prompt attention.

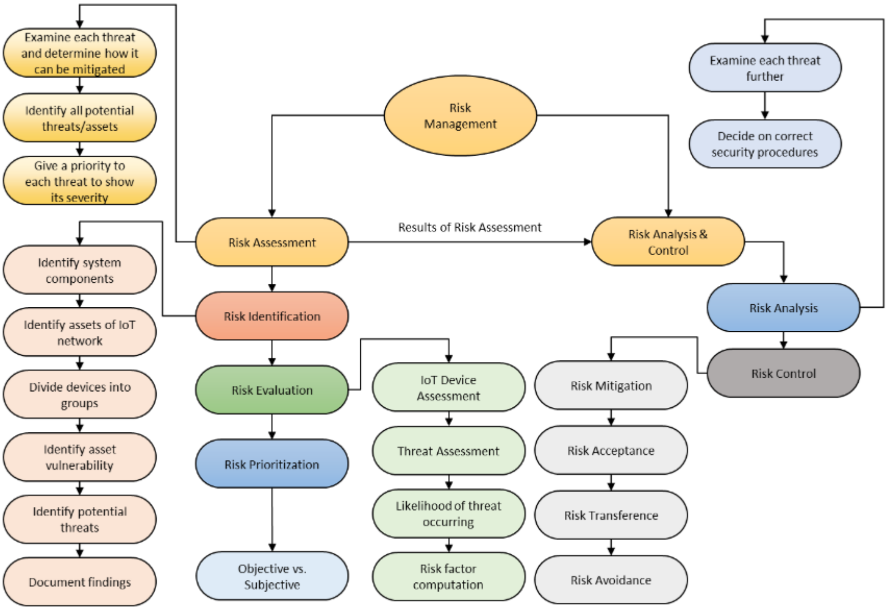
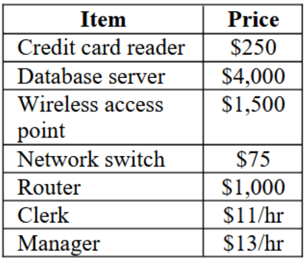

Examine each threat and determine how it can be mitigated
Identify all potential threats/assets
Give a priority to each threat to show its severity
Risk Management
Examine each threat further
Decide on correct security procedures
Results of Risk Assessment
Risk Assessment
Risk Analysis & Control
Identify system components
Risk Identification
Risk Analysis
Identify assets of IoT network
Risk Control
Risk Evaluation
IoT Device Assessment
Risk Mitigation
Divide devices into groups
Identify asset vulnerability
Risk Prioritization
Threat Assessment
Risk Acceptance
Identify potential threats
Likelihood of threat occurring
Risk Transference
Document findings
Objective vs. Subjective
Risk factor computation
Risk Avoidance

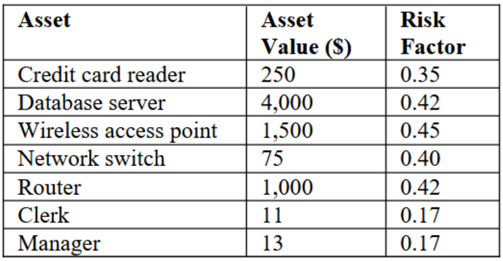
Figure 2: Risk Management Framework for IoT Network

Table 1: Items and Prices



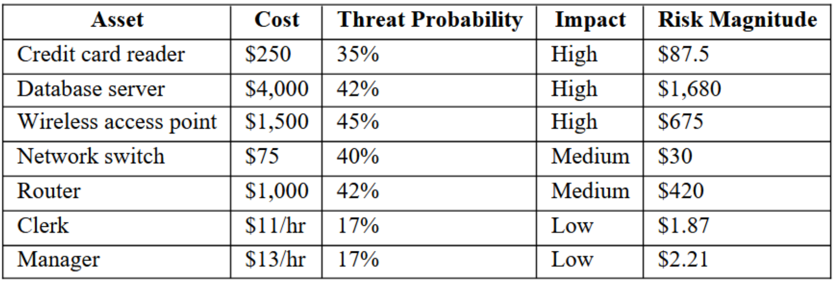
Item	Price
Credit card reader	$250
Database server	$4,000
Wireless access point	$1,500
Network switch	$75
Router	$1,000
Clerk	$11/hr
Manager	$13/hr

Table 2: Assets and Associated Risk



Asset	Asset Value ($)	Risk Factor
Credit card reader	250	0.35
Database server	4,000	0.42
Wireless access point	1,500	0.45
Network switch	75	0.40
Router	1,000	0.42
Clerk	11	0.17
Manager	13	0.17

Table 3: Risk Management Calculation for Assets



Asset	Cost	Threat Probability	Impact	Risk Magnitude
Credit card reader	$250	35%	High	$87.5
Database server	$4,000	42%	High	$1,680
Wireless access point	$1,500	45%	High	$675
Network switch	$75	40%	Medium	$30
Router	$1,000	42%	Medium	$420
Clerk	$11/hr	17%	Low	$1.87
Manager	$13/hr	17%	Low	$2.21

3.3 Risk Prioritization

The prioritization of risks holds significant importance within the risk management process as it allows for effective allocation of time and resources, addressing the most critical risks in a timely manner. Based on our research and an objective assessment, it is recommended that Dollar General prioritize their assets in the following order: database servers as the foremost priority, followed by wireless access points. The third priority should be given to routers, while credit card readers should be addressed as the fourth priority and the network switch as the fifth. Lastly, but not insignificantly, people should be considered. From a subjective standpoint, Dollar General should primarily focus on assets that directly impact their business operations, thus giving priority to credit card readers, database servers, and people as the top three concerns.

3.4 Risk Analysis

Risk analysis is a crucial step in which each threat is thoroughly examined, and appropriate security measures are determined. The probability of skimming devices occurring is assessed at 35%, indicating a moderate threat level. Although workers can potentially identify and prevent skimming device attachment, there remains a possibility of occurrence under specific circumstances. Cyber threats such as SQL injection, data breaches, piggybacking, and DHCP spoofing depend on the location and population of the Dollar General store. If the store is situated in an urban area, the probability of these threats increases due to continuous exposure to diverse networks and individuals. Lastly, estimating the threat of social engineering can be challenging as individuals' awareness of these risks varies. Proper training plays a significant role in effectively mitigating this threat.

3.5 Risk Control

The final step of the framework is paramount to its overall effectiveness. This stage involves the development, implementation, and monitoring of security policies aimed at reducing or eliminating risks. To commence this process, Dollar General should establish comprehensive physical and cyber security policies that raise awareness among employees regarding potential threats and educate them on appropriate response procedures. In addressing cyber threats, robust protection measures must be implemented for all electronic assets identified earlier. This entails regularly updating devices and ensuring they run the latest software versions. Additionally, the use of antivirus software, intrusion detection systems (IDS), and intrusion prevention systems (IPS) should be enforced continuously to maintain operational integrity. Also, Dollar General should maintain up-to-date backups of all stored data as a contingency plan in case of a successful attack.

4 Business Impact Analysis

The BIA operates under the assumption that the security controls established during the risk management process have been circumvented or have proven ineffective in preventing an attack, resulting in a successful breach. The BIA model utilizes the outputs generated by the risk framework as its inputs and builds upon the progress made in the risk management process. Although there is some overlap in the inputs and outputs, certain variations exist. The following steps outline the requirements for the BIA model.

4.1 Step 1: Identify, Analyze, and Prioritize Critical Business Processes and Recovery Criticality

In light of a security breach that has bypassed the system's protective measures, our focus shifts to examining the various business processes conducted at a Dollar General site. By identifying and analyzing these processes, we can determine which ones are vital for the development of a comprehensive business contingency plan. Among the key processes are security camera surveillance, credit card transactions, inventory management, storage server operations, and network connectivity maintenance. After compiling the list of processes, we proceed with a thorough analysis to ascertain the criticality of each process. From this analysis, we have identified the following essential processes: checkout process, inventory management, data storage, security system operations, and network connectivity maintenance. We have established tolerance parameters for each business function to guide our contingency planning, as shown in Fig. 3 to Fig. 7. The figures presented in ascending order represent the recovery criticality of each process, with Fig. 3 being the most critical and Fig. 7 being the least critical. This prioritization enables us to determine which processes require immediate attention and remediation to ensure uninterrupted business operations.

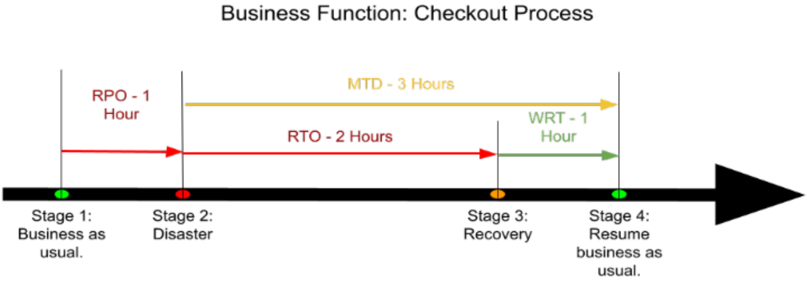

Business Function: Checkout Process
RPO - 1 Hour
MTD - 3 Hours
WRT - 1 Hour
RTO - 2 Hours
Stage 1: Business as usual.
Stage 2: Disaster
Stage 3: Recovery
Stage 4: Resume business as usual.

Figure 3: Tolerance Parameters for the Checkout Process Business Function

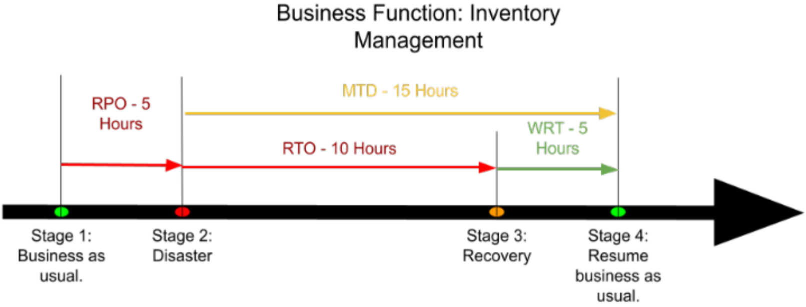

Business Function: Inventory Management
RPO - 5 Hours
MTD - 15 Hours
WRT - 5 Hours
RTO - 10 Hours
Stage 1: Business as usual.
Stage 2: Disaster
Stage 3: Recovery
Stage 4: Resume business as usual.

Figure 4: Tolerance Parameters for the Inventory Management Business Function

4.2 Step 2: Identify Resource Requirements

In this phase, we are tasked with identifying the resources needed to recover the prioritized processes and their associated assets. The checkout process relies heavily on various resources, including the card reader, computers/checkout terminals, server connectivity, and handheld devices. These connections are vital for the smooth functioning of all business functions. The inventory management system primarily relies on handheld devices and the server, which facilitate the distribution, organization, and tracking of materials, pricing, avail-

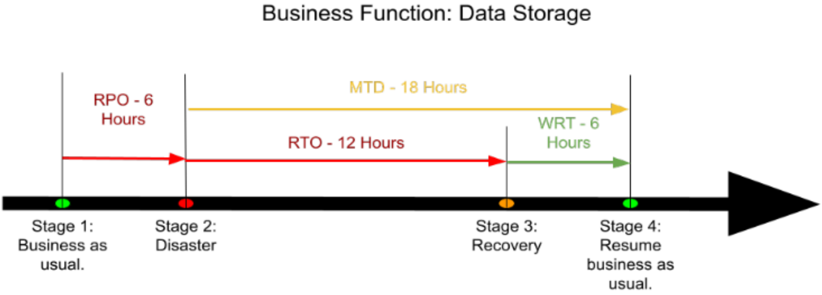

Business Function: Data Storage
RPO - 6 Hours
MTD - 18 Hours
WRT - 6 Hours
RTO - 12 Hours
Stage 1: Business as usual.
Stage 2: Disaster
Stage 3: Recovery
Stage 4: Resume business as usual.

Figure 5: Tolerance Parameters for the Data Storage Business Function

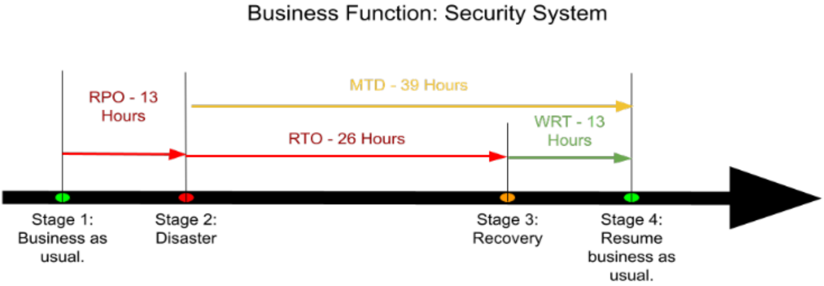

Business Function: Security System
RPO - 13 Hours
MTD - 39 Hours
WRT - 13 Hours
RTO - 26 Hours
Stage 1: Business as usual.
Stage 2: Disaster
Stage 3: Recovery
Stage 4: Resume business as usual.

Figure 6: Tolerance Parameters for the Security System Business Function

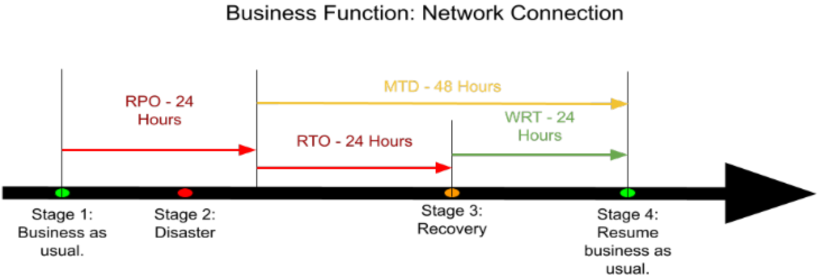

Business Function: Network Connection
RPO - 24 Hours
MTD - 48 Hours
WRT - 24 Hours
RTO - 24 Hours
Stage 1: Business as usual.
Stage 2: Disaster
Stage 3: Recovery
Stage 4: Resume business as usual.

Figure 7: Tolerance Parameters for the Network Connection Business Function

ability, and product details. The data storage function primarily depends on the data server, while the handheld devices and security cameras contribute data to the server but are not essential for its core operation. The security system requires the camera, network connection, and data storage functions, but any disruptions to this function do not directly impact daily business operations. Consequently, it ranks lower on the criticality scale. Lastly, the network connection relies on the router and the internet service provider connection. Although the business stores data locally, a live network connection is crucial for data dissemination to external entities or the security system.

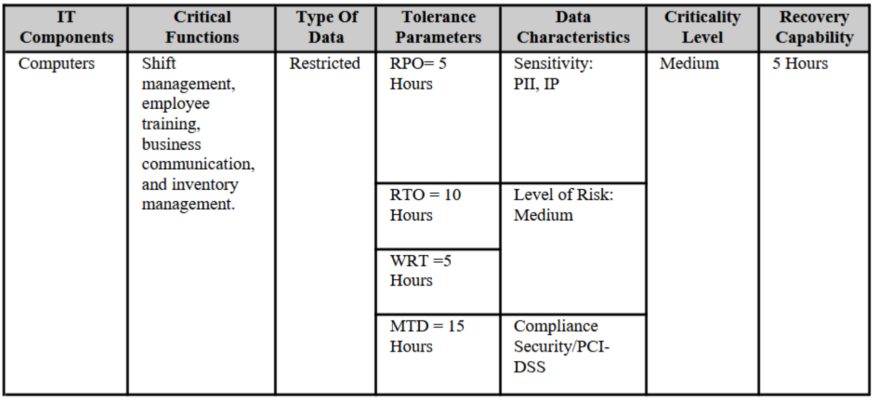
4.3 Step 3: Identify Recovery Priorities

In the final step of the BIA model, the prioritization of resources associated with mission/business processes is undertaken to determine the optimal recovery sequence. This process aims to enhance our understanding of which elements should be restored first. Based on our findings from step 1, the checkout process emerges as the top recovery priority as it aligns with the core objective of the business, involving the sale of goods and payment transactions. Following closely, the inventory management system ranks as the second recovery priority, given its direct support to the primary business function by effectively tracking item details, locations, and related information. Subsequently, the data storage process assumes the third most critical recovery priority, considering the crucial role of stored data in determining inventory needs, tracking sales, and providing comprehensive sales information. The security system is assigned the second-to-last recovery priority, with its significance elevated during physical emergencies despite its reliance on network connectivity. Lastly, the network connection is assigned the lowest recovery priority, as the business can still function fundamentally without an internet connection. Sales can be conducted offline, and upon reestablishing the connection during the work recovery time (WRT) phase, physical sales can be accurately recorded and updated.

5 Data Protection and Recovery Plan

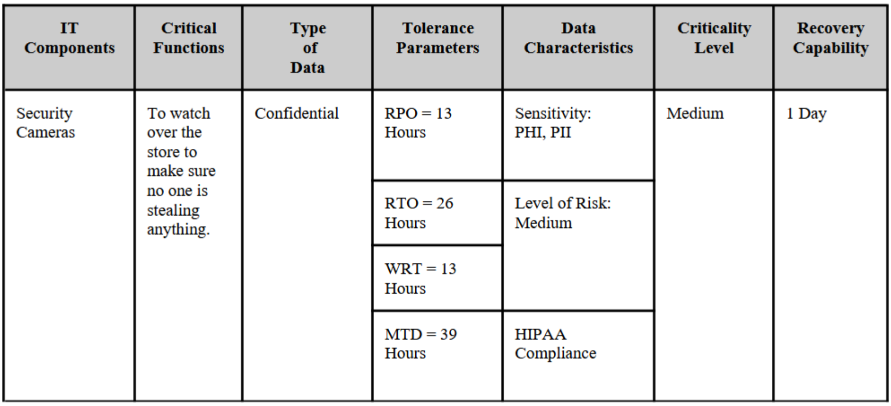
In Tables 4-7 provided below, a comprehensive breakdown of each IoT device deployed within our Dollar General store is presented. The critical functions performed by each device in our operational processes are identified based on inputs from both the risk management framework and the BIA. The tolerance parameters for each device are derived from the BIA conducted in the preceding step. Considering the 'type of data' specified in the tables, the severity and nature of the data collected and stored are taken into account. For hand-held devices and security cameras, the data is categorized as confidential, requiring specific authorization and clearance for access. Hand-held data pertains to inventory details, prices, and locations, accessible on a 'need to know' basis.

Table 4: Computers: Device Data Protection and Recovery Analysis



IT Components	Critical Functions	Type Of Data	Tolerance Parameters	Data Characteristics	Criticality Level	Recovery Capability
Computers	Shift management, employee training, business communication, and inventory management.	Restricted	RPO= 5 Hours	Sensitivity: PII, IP	Medium	5 Hours
			RTO = 10 Hours	Level of Risk: Medium		
			WRT =5 Hours			
			MTD = 15 Hours	Compliance Security/PCI-DSS		

Table 5: Security Cameras: Device Data Protection and Recovery Analysis



IT Components	Critical Functions	Type of Data	Tolerance Parameters	Data Characteristics	Criticality Level	Recovery Capability
Security Cameras	To watch over the store to make sure no one is stealing anything.	Confidential	RPO = 13 Hours	Sensitivity: PHI, PII	Medium	1 Day
			RTO = 26 Hours	Level of Risk: Medium		
			WRT = 13 Hours			
			MTD = 39 Hours	HIPAA Compliance		

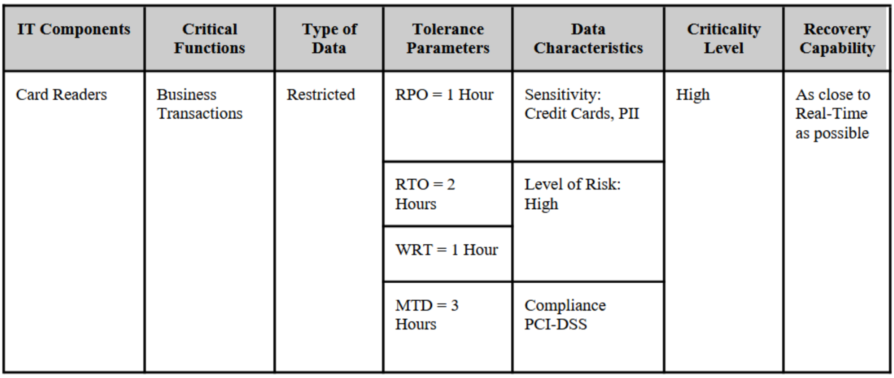
Similarly, security camera data is deemed confidential and accessed primarily during physical incidents or disasters to serve as evidence for property damage, theft, or harm to individuals or employees. Security and emergency services may require the recorded data for confirmation or analysis purposes.

The sensitivity level of each device encompasses personal health information (PHI), personally identifiable information (PII), credit card information, and IP address sensitivity. Compliance requirements for our IoT devices in Dollar General encompass HIPAA and PCI-DSS, focusing on credit card information and surveillance cameras. Credit card readers exhibit high risk and criticality levels due to their integral role in daily operations and the sensitive data they handle. Real-time recovery capabilities are desired for these devices. Security cameras and computer systems carry medium risk and criticality levels since the business can still function adequately without them, albeit with certain limitations and unattainable objectives. Recovery capabilities for these devices range from 5 to 24 hours. Lastly, hand-held devices are considered low-risk and non-critical as the business can operate using manual inventory checks and updates in their absence. Recovery capabilities for these devices span from 1 day to 1 week.

5.1 Plan for Data Protection

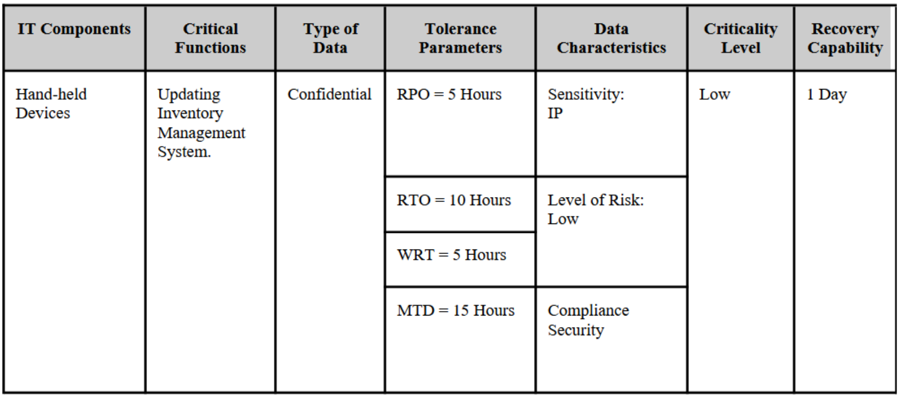
Table 8 outlines establishing backup strategies for our IoT devices at the Dollar General Store. This table guides us in determining the appropriate backup strategy for each device, including the frequency of backups.

Table 6: Card Readers: Device Data Protection and Recovery Analysis



IT Components	Critical Functions	Type of Data	Tolerance Parameters	Data Characteristics	Criticality Level	Recovery Capability
Card Readers	Business Transactions	Restricted	RPO = 1 Hour	Sensitivity: Credit Cards, PII	High	As close to Real-Time as possible
			RTO = 2 Hours	Level of Risk: High		
			WRT = 1 Hour			
			MTD = 3 Hours	Compliance PCI-DSS		

Table 7: Hand-held Devices: Device Data Protection and Recovery Analysis



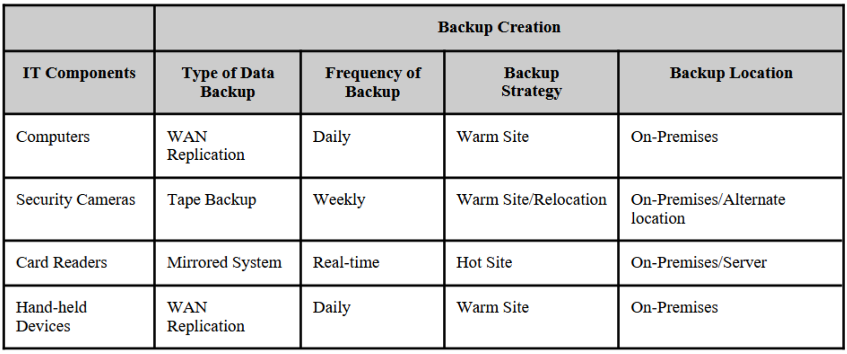
IT Components	Critical Functions	Type of Data	Tolerance Parameters	Data Characteristics	Criticality Level	Recovery Capability
Hand-held Devices	Updating Inventory Management System.	Confidential	RPO = 5 Hours	Sensitivity: IP	Low	1 Day
			RTO = 10 Hours	Level of Risk: Low		
			WRT = 5 Hours			
			MTD = 15 Hours	Compliance Security		

Factors such as the type of site required and its location are considered when making these decisions. By completing all the necessary fields in this framework, we will be able to ascertain the backup creation requirements for each IoT device and subsequently develop a comprehensive backup plan.

5.2 Plan for Data Protection

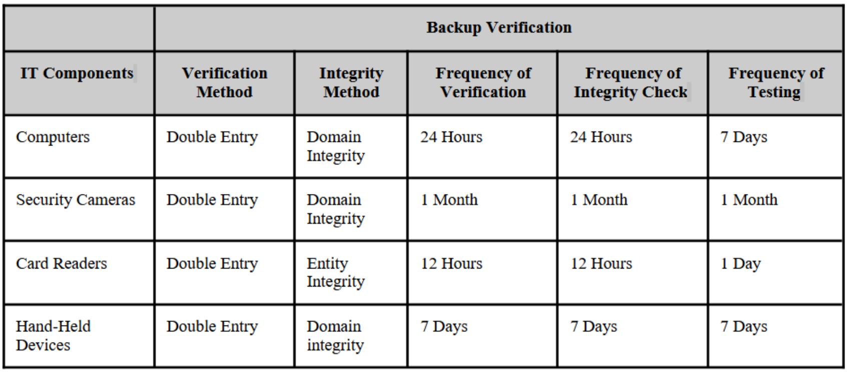
Table 8 outlines establishing backup strategies for our IoT devices at the Dollar General Store. This table guides us in determining the appropriate backup strategy for each device, including the frequency of backups. Factors such as the type of site required and its location are considered when making these decisions. By completing all the necessary fields in this framework, we will be able to ascertain the backup creation requirements for each IoT device and subsequently develop a comprehensive backup plan.

Table 8: IoT Device Data Protection and Recovery Plan



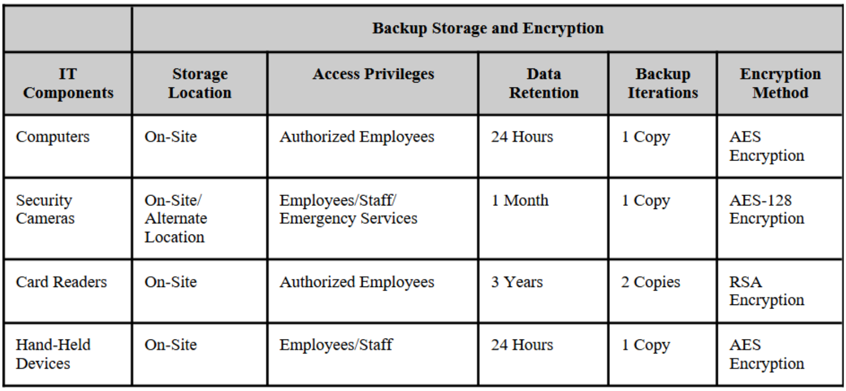
	Backup Creation			
IT Components	**Type of Data Backup**	**Frequency of Backup**	**Backup Strategy**	**Backup Location**
Computers	WAN Replication	Daily	Warm Site	On-Premises
Security Cameras	Tape Backup	Weekly	Warm Site/Relocation	On-Premises/Alternate location
Card Readers	Mirrored System	Real-time	Hot Site	On-Premises/Server
Hand-held Devices	WAN Replication	Daily	Warm Site	On-Premises

Table 9: IoT Device Data Protection and Recovery Plan



	Backup Verification				
IT Components	**Verification Method**	**Integrity Method**	**Frequency of Verification**	**Frequency of Integrity Check**	**Frequency of Testing**
Computers	Double Entry	Domain Integrity	24 Hours	24 Hours	7 Days
Security Cameras	Double Entry	Domain Integrity	1 Month	1 Month	1 Month
Card Readers	Double Entry	Entity Integrity	12 Hours	12 Hours	1 Day
Hand-Held Devices	Double Entry	Domain integrity	7 Days	7 Days	7 Days

Table 10: IoT Device Data Protection and Recovery Plan



	Backup Storage and Encryption				
IT Components	Storage Location	Access Privileges	Data Retention	Backup Iterations	Encryption Method
Computers	On-Site	Authorized Employees	24 Hours	1 Copy	AES Encryption
Security Cameras	On-Site/ Alternate Location	Employees/Staff/ Emergency Services	1 Month	1 Copy	AES-128 Encryption
Card Readers	On-Site	Authorized Employees	3 Years	2 Copies	RSA Encryption
Hand-Held Devices	On-Site	Employees/Staff	24 Hours	1 Copy	AES Encryption

Table 9 refers to assigning a backup verification plan to each IoT device within the framework. This involves determining the specific methods to be employed for verification and ensuring data integrity. Additionally, the frequency of utilizing each verification method needs to be established, and the frequency of testing between verification and integrity checks. The primary purpose of backup verification is to ensure the accuracy and reliability of backups for all IoT devices. Table 10 is dedicated to establishing the backup storage and encryption protocols to be implemented within the Dollar General Store. This framework will determine the designated IoT device that will serve as the storage location for all IoT devices. Additionally, it will outline the authorization process for individuals within the company who will be granted access to these specific IoT devices. Given the significance of ensuring appropriate privileges for different levels of IoT device security, careful consideration will be given to defining these access rights. The framework will also address Data Retention, specifying the duration for which data will be stored. This entails determining the number of backup iterations required and establishing the encryption methods that will be employed to safeguard the data stored within the designated storage device.

6 Conclusion

In conclusion, the implementation of our Risk Management Framework and the Business Impact Analysis (BIA) at Dollar General has significantly contributed to enhancing the security and resilience of the store's IoT ecosystem. Through the rigorous analysis of data protection and recovery measures for each IoT device, we have established a robust foundation for safeguarding data and ensuring business continuity in the face of potential incidents or disasters. This research highlights the critical importance of proactive risk management in the retail sector and emphasizes the need for comprehensive contingency plans. By diligently adhering to these measures, organizations can minimize the impact of potential disruptions, protect customer information, and maintain the integrity of their operations. Furthermore, this study opens up new avenues for future research. The integration of artificial intelligence (AI) and machine learning (ML) techniques can further enhance the effectiveness and efficiency of risk management practices in IoT environments. By leveraging AI/ML-based schemes, organizations can continuously monitor and adapt their security measures, proactively identifying and mitigating emerging threats. Overall, this research contributes to the evolving field of IoT risk management in the retail sector, paving the way for advanced AI/ML-based solutions and inspiring further exploration in securing IoT ecosystems.

References

[1] Roman Chuprina and Olena Kovalenko. *Artificial Intelligence for Retail in 2022: 12 Real-World Use Cases*. Sept. 2022. URL: `https://insights.sei.cmu.edu/blog/a-new-approach-to-cyber-incident-response/`.

[2] Anne Connell. *A new approach to cyber incident response*. Feb. 2014. URL: `https://insights.sei.cmu.edu/blog/a-new-approach-to-cyber-incident-response/`.

[3] Nomusa Nomhle Dlamini and Kevin Johnston. "The use, benefits and challenges of using the Internet of Things (IoT) in retail businesses: A literature review". In: *2016 international conference on advances in computing and communication engineering (ICACCE)*. IEEE. 2016, pp. 430–436.

[4] Keshav Kaushik and Susheela Dahiya. "Security and privacy in IoT based e-business and retail". In: *2018 International Conference on System Modeling & Advancement in Research Trends (SMART)*. IEEE. 2018, pp. 78–81.

[5] Sandeep Shekhawat. "Use of AI and IoT to make Retail Smarter". In: *2022 3rd International Informatics and Software Engineering Conference (IISEC)*. IEEE. 2022, pp. 1–5.

[6] Gurinder Singh et al. "Companies adoption of IoT for smart retailing in industry 4.0". In: *2020 International Conference on Intelligent Engineering and Management (ICIEM)*. IEEE. 2020, pp. 487–492.

N-Dolphin: A Visualizer for Abstract Substitution-Based Execution*

Alex Reichard[1] and David G. Wonnacott[2]
[1]alex.reichard2@gmail.com

[2]Computer Science Department
Haverford College
Haverford, PA 19041
davew@cs.haverford.edu

Abstract

This paper presents N-Dolphin, a notional-machine visualizer coded in Racket, designed to enhance the learning experience for novice programmers by offering an interactive representation of substitution-based execution of Python code.

N-Dolphin aims to make advanced programming concepts more intuitive and accessible to students. The tool offers a variety of code traversal techniques and novel functionality to help a beginner build a solid foundation for abstract reasoning about code and computation. N-Dolphin provides both substitution-based execution and a "step-over" feature, which are not combined in any readily-available tool, despite each, individually, being valuable for teaching challenging topics like recursion. Furthermore, N-Dolphin provides additional symbolic substitution steps, to provide support for symbolic reasoning about code. Finally, it provides an intuitive user interface for students to interact with code and visualize its execution.

Although N-Dolphin is written in Racket and supports substitution-based execution, it currently presents code in eclectic-but-legal Python.

1 Introduction

Beginning programmers need to create mental models of the execution of code they build, but doing so can be challenging. DuBoulay uses the term *notional machine* for these models [1], and Sorva argues that "instructors should acknowledge the notional machine as an explicit learning objective and address it in teaching" [8], so that students are not left to form their own partially-correct models that must then be partially unlearned.

We have been exploring the interplay between notional machine details, visualization, and pedagogy. It is our hope that an appropriate combination of pedagogy and visualization software will help to address some long-standing challenges in the introductory computer science curriculum, in particular, improving the understand of *abstraction*, ability to use *recursion*, and appreciation of *mathematical reasoning*, in beginning students.

To leverage student enthusiasm for focusing on code rather than proofs, we frame all of these activities in terms of a pure-functional subset of Python. The N-Dolphin examples in this paper involve numbers, though the core of N-Dolphin could also handle pure-functional operations on strings, lists, etc., and could, in principle, be modified to allow other languages or other sets of Python idioms.

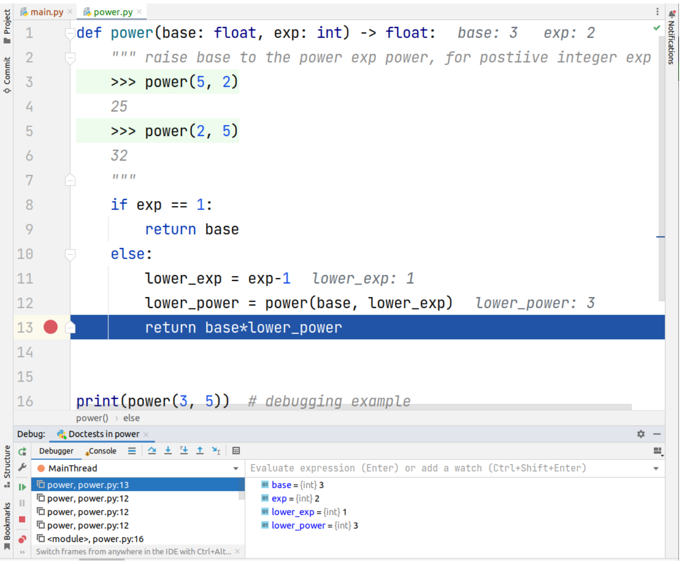
1.1 Visualizers and Notional Machines

Most experienced programmers are familiar with a some sort of automated debugging tool, such as the PyCharm debugger shown in Figure 1. A debugger provides a simplified visualization of the internal machine state. Note that, depending on language and options chosen, the visualization may hide any number of details of the execution:

- Language-implementation overheads may be omitted, e.g., the computer memory holds a return address, as well as parameters and variables, when running the recursive call to power of Figure 1.

- Optimization steps may be ignored/undone in some debuggers, e.g., if we were to debug a C/C++ version of Figure 1, the debugger could provide a less-confusing view by including the values of both local variables simultaneously, as in Figure 1, regardless of whether or not the machine actually stores them in memory simultaneously.

- Data-structure implementation details may be omitted, e.g., a Python list of lists might be printed without any indication about whether identical sub-lists are copies or actually references to the same list, as in Figure 2.

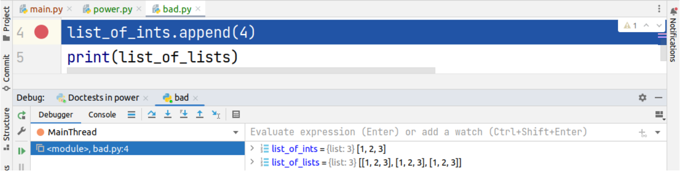


```
def power(base: float, exp: int) -> float:   base: 3   exp: 2
    """ raise base to the power exp power, for postiive integer exp
    >>> power(5, 2)
    25
    >>> power(2, 5)
    32
    """
    if exp == 1:
        return base
    else:
        lower_exp = exp-1   lower_exp: 1
        lower_power = power(base, lower_exp)   lower_power: 3
        return base*lower_power


print(power(3, 5))  # debugging example
```

```
MainThread
power, power.py:13
power, power.py:12
power, power.py:12
power, power.py:12
<module>, power.py:16
Switch frames from anywhere in the IDE with Ctrl+Alt...
```

```
Evaluate expression (Enter) or add a watch (Ctrl+Shift+Enter)
base = {int} 3
exp = {int} 2
lower_exp = {int} 1
lower_power = {int} 3
```

Figure 1: Debugging a Recursive Function in PyCharm.



```
4  list_of_ints.append(4)
5  print(list_of_lists)
```

```
MainThread
<module>, bad.py:4
```

```
Evaluate expression (Enter) or add a watch (Ctrl+Shift+Enter)
> list_of_ints = {list: 3} [1, 2, 3]
> list_of_lists = {list: 3} [[1, 2, 3], [1, 2, 3], [1, 2, 3]]
```

Figure 2: Debugging a List with Sub-lists in PyCharm.

Professional programmers need to be familiar with the strengths and limitations of such tools, but a novice may take this visualization as a description of what is "actually happening" as the code runs. Visualizations such as Figures 1 and 2 may help beginning students understand things that are highlighted in the debugger, e.g., how recursive functions manage to arrive at the right result, but confuse them about things that are simplified, e.g., references to, vs. copies of, a list. Although Figure 2 shows the complete set of variables and values for that mini-example, it provides us no way to know that Line 5 will print `[[1, 2, 3, 4], [1, 2, 3], [1, 2, 3, 4]]`, and, by itself, no way to understand that result when it is printed. Thus, instructors much select such visualizers carefully, exploring the code from Figure 2 with a visualizer like Python Tutor [6] or hand-drawn "box and arrow" diagrams rather than PyCharm.

Whether used by a novice or expert, a notional machine or visualizer should draw a user's attention to something that may not have been obvious when simply looking at a piece of code, and, when debugging, on some aspect that is relevant to the current problem. Visualizers can focus on the specific structures of a language, the processes in the machines controlled by the language (e.g., Python storing all of its values as objects), or how the two intersect [5].

Different visualizations can not only help students understand a variety of different kinds of code, but also provide a variety of ways to understand a single piece of code. The PyCharm debugger and Python Tutor present variations on the "dictionary notional machine", in which program execution is understood as a process in which line-by-line execution of code updates updating a set of mappings from names to object values (i.e., a dictionary). In contrast, the steps of the "substitution notional machine" [9], resemble the substitution rules used is high-school algebra, e.g., replacing a variable name with its definition. While the substitution notional machine is not appropriate for code in which variables or objects change, it can help beginners understand recursive code like that of Figure 1, as discussed by Tunnel-Wilson, Fisler, and Krishnamurthi [9]. This work uses the Racket language, and uses the "stepper" tool of the DrRacket programming environment[3] to provide a visualization for the substitution notional machine.

1.2 Our Experiences and Goals

The substitution notional machine is naturally suited to pure-functional code, and Racket inherits a tradition of pure-functional coding. However, Racket may not be an optimal first language for every student (e.g., those motivated by data science) or at every institution. We believe that abstract thinking about code and computation is an important element of CS education regardless of language. Furthermore, we have found the substitution notional machine lets us introduce a number of abstract and symbolic reasoning skills in coding classes, rather than reserving them for a separate "Discrete Math" course. Thus, we are focused on using substitution on a pure-functional subset of Python. Our prior exploration of this pedagogy, with pencil-and-paper exercises and non-standard subsets of Python, convinced us that we need *automated tools* that show only *legal* Python. Our next steps focus on using such tools with pre-loaded examples (rather than arbitrary student code) and exploring various idioms within legal Python.

The DrRacket stepper also lacks one feature that is commonplace in professional debuggers: a "step-over" button that gets from a *call* to a function to the *returned value* in a single click. While this may seem like a minor oversight — in small student programs, one could just hit "step" a bunch of times, to reach the end of the function — we find it a significant barrier to teaching students to reason *abstractly* about code. This is particularly important in the teaching of recursion, where our experience [7] matches significant anecdotal evidence from other instructors: students may need to see both *how recursion works*, and also how to *reason about recursion* by "staying at the top level", asking "If all that execution produces the correct answer for the recursive call, what does that mean about the result I'll return as my final answer?". The *workings* of recursion are emphasized by the "step" button, which shows that the computer can get through all those steps and back to the original call, but the latter is highlighted by using "step-over" to get directly from call to result.

Our experience indicates that we can introduce beginners to these two views of code relatively early in their experience as coders. Furthermore, experience discussing both "*how does it work*" and "*what does it produce*", seems to provide a good foundation upon which to build an abstraction-based appreciation for recursion. Thus, we wish to support visualization with both "step" and "step-over" functionality. But, this feature is not available in the DrRacket stepper or Python Tutor.

To support our teaching with substitutional as well as dictionary notional machines [4], we are developing our own visualizers that allow "step over" (substituting the result of a call for the call expression), as well as a variety of other substitution rules that we hope will help us strengthen the connection to our discrete mathematics course. Our first exploration used *Orc*^{2}*a* [2], which is written in JavaScript, and installation requires considerable configuration, and, thus, potential frustration. More recently, we have added *N-Dolphin* (this name arises because dolphins are smaller than orcas, and because N-Dolphin, like Orc^{2}a, lets us execute a function for a symbolic parameter like N rather than a constant like 5).

N-Dolphin and Orc^{2}a both support a wide variety of substitution rules, including step-over and others detailed below. They both currently display code in a pure-functional subset of Python in which all program elements are expressions rather than statements. Specifically, `if` is always displayed as an expression rather than a statement, and local variables and functions are both introduced via `lambda`. Contrast the idiomatic code in Figure 1 with the definition of `power` in the upper left of Figure 3 for an example. Both figures show legal Python code, and we often introduce the expression forms of both `if` and `lambda` in our first semester, with a focus on idiomatic usage (i.e., for one-line expressions). The use of this notation in our visualizers pushes students to have a *reading* knowledge of these expressions in larger contexts, though we do not ask them to *write* code in this form.

2 N-Dolphin

Leveraging the power of abstraction through Notional Machines can enhance the learning experience for novice programming students. By presenting the run-time processes as visual representations, these Notional Machines help students grasp coding concepts in a more intuitive and accessible way. The remainder of this thesis focuses on applying N-dolphin, which is coded into Racket to act as a notional machine visualizer. This project combines many of the pedagogical ideas discussed into one Substitutional Notional Machine. The purpose is to gain a better understanding of how different techniques can help to aid in specific use cases for students and broaden the scope of knowledge surrounding true pedagogical Notional Machines. The following are the capabilities of the aplication sofar.

2.1 N-Dolphin Code Traversal Techniques

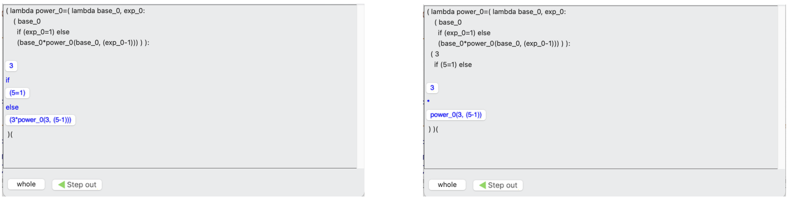
While moving through the code, the user can choose where to apply a substitution, and control the level of abstraction when stepping, with the following buttons.

- **Step-in function:** This is the native traversal technique in the DrRacket language. It enables the user to execute code line-by-line, venturing into any functions within the current scope. This feature facilitates exploration of the function's mechanics, tracing the program's flow as it runs. It sheds light on variable and procedure values, tracks the program's progress, and clarifies otherwise obscure aspects of the code. It steps inward, executing every directive of the code when reached.

- **Step out function:** Takes a step out, moving up one level.

- **Step-over function:** Allows execution of a function without entering any internal procedures. The debugger treats the function as a singular entity and progresses to the following code within the current scope. It internally repeats substitutions without displaying results until it reduces to its singular value. This feature is beneficial when the internal execution of a particular function isn't of interest, and the focus is on its impact on the rest of the code.

- **Order Manipulation:** Empowers the user to manipulate how code segments are managed. The user can step into functions and run the step-in and step-over functions on internal code segments in an order distinct from the machine's predetermined order. Currently, the user steps further into children of a function using buttons, or stepping out a layer using the "step out" button. Figure 3 illustrates the process in



```
( lambda power_0=( lambda base_0, exp_0:
   ( base_0
    if (exp_0=1) else
    (base_0*power_0(base_0, (exp_0-1))) ) ):


3
if
(5=1)
else
(3*power_0(3, (5-1)))
)(
```

selected code from upcoming Figure 4

```
( lambda power_0=( lambda base_0, exp_0:
   ( base_0
    if (exp_0=1) else
    (base_0*power_0(base_0, (exp_0-1))) ) ):
 ( 3
  if (5=1) else


3
*
power_0(3, (5-1))
) )(
```

step-in using buttons to manipulate function

Figure 3: Order Manipulation

a simple example, where instead of going to the condition (5=1), the buttons allow the user to choose which part of the code is selected for manipulation. This is integral to our ongoing work in tying it into discrete math.

- **Go Back a Step:** This returns the user to the configuration one step prior.

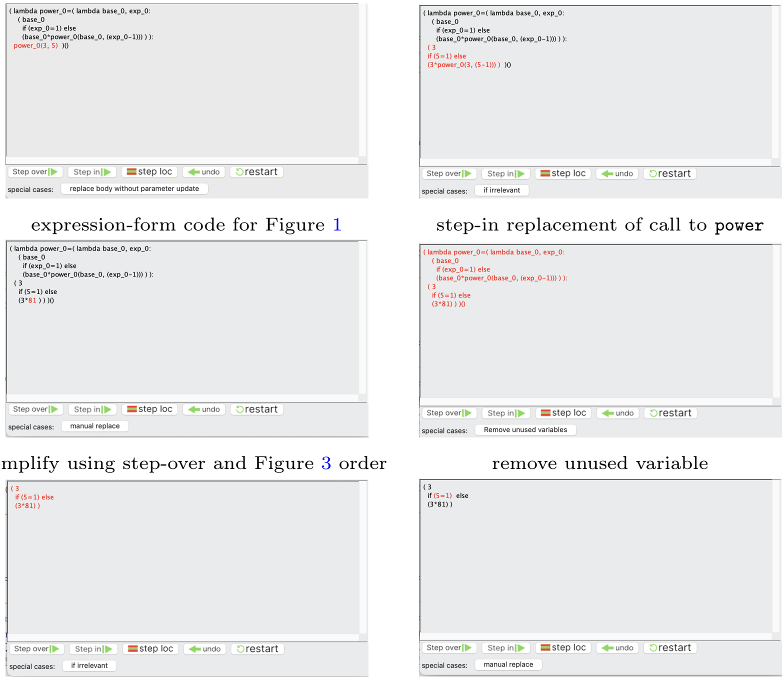
2.2 Non-native step functionality

While these techniques are not the primary focus of this paper, N-dolphin encompasses functionality that extends beyond the usual model of execution, and supports even more advanced symbolic program transformation. These are designed to help students as they advance and begin to explore reasoning about abstract properties of code; for details, see [10].

To support this pedagogy, N-dolphin also presents buttons for the following when suitable for the code context:

- **Symbolic Function Call:** This grants a user the capability to invoke a function without initially resolving the function parameter values completely. This is useful for attributing variable names rather than values to a function for the purpose of demonstrating its correctness.

- **Symbolic Term Replacement:** This empowers users to manually substitute a term with an equivalent expression using symbolic math, an algebraic manipulation of mathematical expressions rather than numerical calculations. For instance, in the context of N-dolphin, an expression such as $(a + b)$ can be manually transmuted into $(2 * a)$, given $(a = b)$. Equivalence is validated using the "rosette" package in DrRacket, and the terms under consideration (e.g., $a = b$) are extracted from any encompassing if-statements.



```
( lambda power_0=( lambda base_0, exp_0:
   ( base_0
    if (exp_0=1) else
    (base_0*power_0(base_0, (exp_0-1))) ) ):
 power_0(3, 5)  )()
```

special cases: replace body without parameter update

expression-form code for Figure 1

```
( lambda power_0=( lambda base_0, exp_0:
   ( base_0
    if (exp_0=1) else
    (base_0*power_0(base_0, (exp_0-1))) ) ):
 ( 3
 if (5=1) else
 (3*power_0(3, (5-1))) )  )()
```

special cases: if irrelevant

step-in replacement of call to `power`

```
( lambda power_0=( lambda base_0, exp_0:
   ( base_0
    if (exp_0=1) else
    (base_0*power_0(base_0, (exp_0-1))) ) ):
 ( 3
   if (5=1) else
   (3*81 ) ) )()
```

special cases: manual replace

simplify using step-over and Figure 3 order

```
( lambda power_0=( lambda base_0, exp_0:
   ( base_0
    if (exp_0=1) else
    (base_0*power_0(base_0, (exp_0-1))) ) ):
 ( 3
   if (5=1) else
   (3*81) ) )()
```

special cases: Remove unused variables

remove unused variable

```
( 3
  if (5=1) else
  (3*81) )
```

special cases: if irrelevant

unused let removed

```
( 3
  if (5=1)  else
  (3*81) )
```

special cases: manual replace

step-in moves to next rule

Figure 4: Stepping In and Stepping Over in N-Dolphin

- **If Irrelevant:** This substitutes an if-statement with only the true branch, if both the if and else statements are equivalent. This lets us frame proof-by-cases as code execution.
- **Name-to-spec** and **Name-to-spec-simpler:** These allow substitution of a value specified by a trusted function's postcondition, to allow abstract use of library functions for which we have no code, or to allow inductive reasoning as discussed in [10].

2.3 User Interface

Figure 4 demonstrates the code stepping and traversal aspects of N-Dolphin, on a variant of the Python code from Figure 1. After loading a file, we intend to have N-Dolphin first present the code as idiomatic Python, as in Figure 1, and then convert to all code to pure Python expressions, e.g., turning an if statement into an if expression. Earlier exploration of curricula involved doing substitution on a mix of statements and expressions, which seemed to be even more confusing than having multiple notations for things like `if`, though we may explore this again once we have better tool support. Note that we do not expect students to *write* code in this form, only to see it as equivalent when they explore substitution-based execution.

The code in Figure 4 also shows the form after we have used substitution to remove the local variables `lower_exp` and `lower_power`, which are useful for making a standard debugger retain the recursive parameter and result, but not needed in the context of N-Dolphin.

3 Conclusions and Future Work

Notional machines can be useful as learning tool. Since, by nature, they abstract away many detail of actual physical computing devices, they highlight certain aspects of computation, while downplaying others. Substitution machines can play a critical role in the learning environment, supporting reasoning about recursion, as discussed in [9]. Students also value exposure to abstract thinking about recursion [7].

Unfortunately, no readily-available tool supports abstract reasoning via a "step-over" functionality for a substitution-based machine. Thus, we are exploring this combination both in Orc^{2}a and by developing N-dolphin. To further support abstract reasoning about code, these tools both allow a variety of symbolic substitutions. The use of an automatic visualizer/stepper should greatly reduce the barriers to learning these more advanced approaches to reasoning about code, by automatically handling the details such as

- tracing of functions or recursive functions at a user-controlled level
- automatically renaming variables when needed
- refactoring code

We believe these tools will help students explore a variety of ways of thinking about code execution and appreciate the relevance of abstract/symbolic thinking to code, in the context of activities that involve tool-supported manipulation of Python code rather than hand-written proofs.

We have not yet been able to try out Orc^{2}a or N-dolphin, but hope to do so in the coming semesters, and compare the results with older exams in which students attempted on-paper substitution.

References

[1] Benedict du Boulay, Tim O'Shea, and John Monk. "The black box inside the glass box: presenting computing concepts to novices". In: *International Journal of Man-Machine Studies* 14.3 (1981), pp. 237–249. ISSN: 0020-7373. DOI: https://doi.org/10.1016/S0020-7373(81)80056-9. URL: https://www.sciencedirect.com/science/article/pii/S0020737381800569.

[2] Jianting Chen et al. "ORC2A: A Proof Assistant for Undergraduate Education". In: *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*. SIGCSE '17. Seattle, Washington, USA: Association for Computing Machinery, 2017, pp. 757–758. ISBN: 9781450346986. DOI: 10.1145/3017680.3022466. URL: https://doi.org/10.1145/3017680.3022466.

[3] John Clements, Matthew Flatt, and Matthias Felleisen. "Modeling an Algebraic Stepper". In: *Proceedings of the 10th European Symposium on Programming Languages and Systems*. ESOP '01. Berlin, Heidelberg: Springer-Verlag, 2001, pp. 320–334. ISBN: 3540418628.

[4] John P. Dougherty and David G. Wonnacott. "Use and Assessment of a Rigorous Approach to CS1". In: *Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education*. SIGCSE '05. St. Louis, Missouri, USA: Association for Computing Machinery, 2005, pp. 251–255. ISBN: 1581139977. DOI: 10.1145/1047344.1047431. URL: https://doi.org/10.1145/1047344.1047431.

[5] Sally Fincher et al. "Notional Machines in Computing Education: The Education of Attention". In: *Proceedings of the Working Group Reports on Innovation and Technology in Computer Science Education*. ITiCSE-WGR '20. Trondheim, Norway: Association for Computing Machinery, 2020, pp. 21–50. ISBN: 9781450382939. DOI: 10.1145/3437800.3439202. URL: https://doi.org/10.1145/3437800.3439202.

[6] Philip Guo. *Python Is Now the Most Popular Introductory Teaching Language at Top U.S. Universities*. en. 2014. URL: https://cacm.acm.org/blogs/blog-cacm/176450-python-is-now-the-most-popular-introductory-teaching-language-at-top-us-universities/fulltext (visited on 10/29/2021).

[7] Gulesh Shukla and David G. Wonnacott. "On Teaching and Testing Recursive Programming". In: *J. Comput. Sci. Coll.* 38.3 (Nov. 2022), pp. 98–106. ISSN: 1937-4771.

[8] Juha Sorva. "Notional Machines and Introductory Programming Education". In: *ACM Trans. Comput. Educ.* 13.2 (July 2013). DOI: 10.1145/2483710.2483713. URL: https://doi.org/10.1145/2483710.2483713.

[9] Preston Tunnell Wilson, Kathi Fisler, and Shriram Krishnamurthi. "Evaluating the Tracing of Recursion in the Substitution Notional Machine". In: *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*. SIGCSE '18. Baltimore, Maryland, USA: Association for Computing Machinery, 2018, pp. 1023–1028. ISBN: 9781450351034. DOI: 10.1145/3159450.3159479. URL: https://doi.org/10.1145/3159450.3159479.

[10] David G. Wonnacott and Peter-Michael Osera. "A Bridge Anchored on Both Sides: Formal Deduction in Introductory CS, and Code Proofs in Discrete Math". In: *CoRR* abs/1907.04134 (2019). arXiv: 1907.04134. URL: http://arxiv.org/abs/1907.04134.

Kira: A Financial Chatbot Using ChatGPT and Data Obfuscation[*]

Telma Búadóttir, Olivia Mascio, Joshua Eckroth
Stetson University
DeLand, FL 32723
tbuadottir,omascio,jeckroth}@stetson.edu

Abstract

Kira is a financial advisory tool that is readily available to assist with a range of financial questions and tasks. Kira communicates strictly through text messages (on a phone). Kira uses OpenAI's ChatGPT to answer questions, the Plaid platform to access users' financial information, and Twilio to send and receive SMS messages. Because ChatGPT is a cloud service, users will likely not be willing to submit financial information to third party. Thus, we developed an algorithm to obfuscate the data before sending it to ChatGPT and reversing the obfuscation in its responses. This tool demonstrates how a useful, complete, and sophisticated application may be built using a suite of tools, including advanced natural language AI as the sole user-facing experience.

1 Introduction

Chatbots and virtual assistants have become increasingly common in banking and finance, providing customers with efficient and personalized services. Most banking chatbots require the user to access the bank's website and complete an authentication each time they need assistance. These applications also mainly focus on answering common questions and often cannot complete tasks such as money transfers.

We developed an assistant that can do both. We call her Kira. Kira is a chatbot that operates exclusively through text messages and is designed to initiate contact after the user completes the initial sign-up process on a website. Following the initial registration, Kira no longer requires login or authentication. Users are authenticated by their phone number. In addition to question-answering, Kira offers a range of other functionalities, such as assistance with loans, budgets, account information, expenses, and account money transfers. This advanced functionality is made possible by integrating technologies including OpenAI's ChatGPT [6] the Plaid platform [9], and Twilio [16]. Many users may object to sending their personal information and account data to OpenAI. To address this issue, we developed an algorithm that obfuscates the user's data before sending it to ChatGPT.

To interact with Kira, a user first establishes a connection between Kira and their bank at kirafinanical.com. They also provide their phone number and thereafter all future interactions are done through text messages. We use the Twilio [16] platform to enable text messaging. When the user sends a message to Kira, we first decide what they are asking by sending their question plus a specific prompt to ChatGPT. Based on ChatGPT's response, we send either obfuscated transactions, obfuscated loan details, or other information so that ChatGPT can produce a meaningful response. We deobfuscate its response to produce a final reply to the user's message. With this workflow, Kira can understand and respond to complex natural language questions and commands, while still preserving the user's financial privacy.

Kira demonstrates the value of integrating multiple tools and APIs together to build advanced applications. Only very recently could something like Kira exist. ChatGPT acts a glue between the SMS capabilities from Twilio and the financial data from Plaid. Previously, another tool such as Rasa [13] would be needed to help developers build a chatbot by providing numerous training examples of user input. With ChatGPT, that work is eliminated. Instead, one must develop "prompts" to guide ChatGPT to make sense of financial data and the user's request. ChatGPT is so powerful and natural in its understanding and responses that we felt comfortable limiting the interface of Kira to just text messages – no tables, plots, buttons, menus, or other elements that one often finds in banking applications. Kira builds on the most recent technologies and shows that AI can simplify the user experience by better understanding the user's intentions.

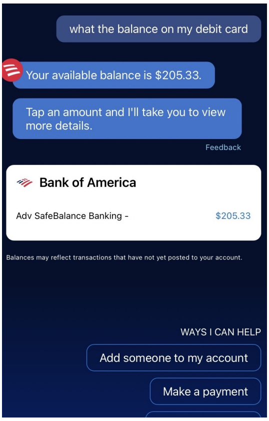
The rest of this work is organized as follows. First, we cover the following background concepts: existing banking chatbots, ChatGPT, Plaid, and Twilio. In Section 3, we detail our implementation, including ChatGPT prompts. Section 4 covers our novel obfuscation/deobfuscation procedures to hide sensitive data from OpenAI. We next discuss the registration website in Section 5, then show example interactions with Kira in Section 6. Finally, we conclude in Section 7.

2 Background

2.1 Chatbots for Banking

In recent years, banking chatbots have become increasingly popular among financial institutions, providing customers with a new way to interact with their banks and manage their finances. These chatbots are powered by artificial intelligence and can provide customers with personalized financial assistance and support, 24/7.

One of the most famous banking chatbots is Erica, developed by Bank of America [3]. Erica (shown in Figure 1 and Figure 2) can answer customers' questions about their accounts, provide balance information, and help them make transfers and payments. Erica is just one example of the many banking chatbots that have emerged in recent years, including Kasisto (used by Chase, TD Bank, and others) in Figure 3, and Amelia (developed by SEB Bank).


what the balance on my debit card

Your available balance is $205.33.

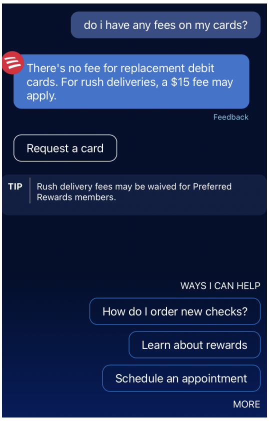
Tap an amount and I'll take you to view more details.

Bank of America

Adv SafeBalance Banking - $205.33

Balances may reflect transactions that have not yet posted to your account.

Figure 1: Erica - Account balance


do i have any fees on my cards?

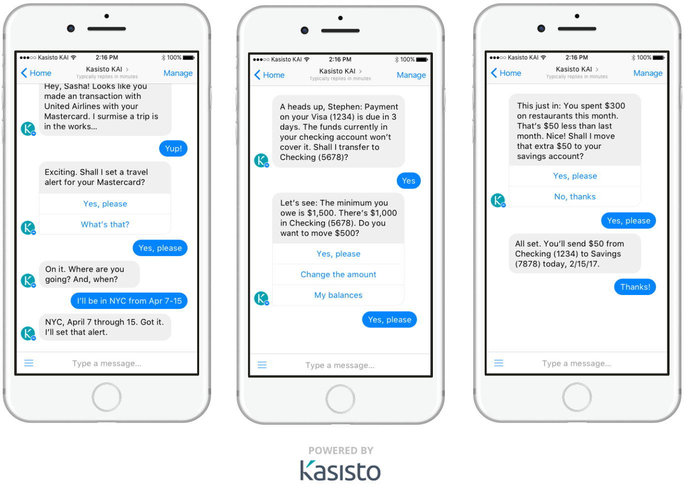
There's no fee for replacement debit cards. For rush deliveries, a $15 fee may apply.

TIP Rush delivery fees may be waived for Preferred Rewards members.

Figure 2: Erica - Fees

2.2 ChatGPT

ChatGPT is a groundbreaking language model developed by OpenAI that has the ability to understand and generate human-like language. The model was originally released in 2020 and was restricted to select partners and organiza-



●●●○○ Kasisto KAI 2:16 PM 100%
Home
Kasisto KAI
Typically replies in minutes
Manage

Hey, Sasha! Looks like you made an transaction with United Airlines with your Mastercard. I surmise a trip is in the works...

Yup!

Exciting. Shall I set a travel alert for your Mastercard?

Yes, please

On it. Where are you going? And, when?

I'll be in NYC from Apr 7-15

NYC, April 7 through 15. Got it. I'll set that alert.

Type a message...
●●●○○ Kasisto KAI 2:16 PM 100%
Home
Kasisto KAI
Typically replies in minutes
Manage

A heads up, Stephen: Payment on your Visa (1234) is due in 3 days. The funds currently in your checking account won't cover it. Shall I transfer to Checking (5678)?

Yes

Let's see: The minimum you owe is $1,500. There's $1,000 in Checking (5678). Do you want to move $500?

Yes, please

This just in: You spent $300 on restaurants this month. That's $50 less than last month. Nice! Shall I move that extra $50 to your savings account?

Yes, please

All set. You'll send $50 from Checking (1234) to Savings (7878) today, 2/15/17.

Thanks!

Figure 3: Kasisto - Example responses

tions. However, in 2022, OpenAI made a significant change to its policy on access to ChatGPT, making the model available to the public [14].

This was a major development for the field of natural language processing, as it allowed more people and organizations to access and benefit from the capabilities of ChatGPT. With the model now available to the public, developers, and businesses of all sizes could use the power of natural language processing to create more engaging and interactive applications and services [14].

To make the technology accessible to a wider audience, OpenAI introduced a usage-based pricing model for the ChatGPT API. This allowed developers to use the model on a pay-as-you-go basis, making it more affordable and accessible for smaller businesses and individual developers [7].

Unlike earlier AI models, which must be trained for specific tasks, ChatGPT is adapted to specific tasks by writing a prompt. For example, if one wishes for ChatGPT to identify the author of a section of text, one simply asks, perhaps also providing a few examples for 'in-situ training.' Thus, one can use the ChatGPT API for any number of disparate tasks by changing the prompt.

2.3 Plaid

Plaid, a financial technology company founded in 2012, offers a platform for building financial services applications. At the core of the company's offerings is the Plaid API, a suite of tools enabling developers to connect their applications with financial institutions and access financial data [11].

The Plaid API equips developers with a robust set of tools for creating innovative financial services applications. Through the API, developers can securely access and analyze financial data from over 11,000 financial institutions across the United States, Canada, and Europe [10]. This data includes account balances, transaction histories, and other financial information that powers a wide variety of financial applications.

The Plaid API has gained popularity for its diverse applications, ranging from personal finance management to lending and investing. The API's ability to securely access financial data from multiple sources has made it an valuable asset for fintech companies and startups seeking to develop innovative financial services.

Beyond providing access to financial data, the Plaid API also features tools for verifying and authenticating user identities, thereby reducing fraud and enhancing security for financial services applications [9].

2.4 Twilio

Twilio is a cloud communication platform that provides developers with a set of APIs for building messaging, voice, and video applications [2]. Twilio's APIs allow developers to add communication functionality to their applications without having to build the infrastructure from scratch [16]. Twilio's APIs support a wide range of communication channels, including Short Message Service (SMS), Multimedia Messaging Service (MMS), voice, video, chat, and email.

Twilio's business model is based on a pay-as-you-go pricing model, where customers pay only for the services they use [2]. Twilio's platform is also highly scalable.

3 Implementation

At a high level, Kira is implemented as follows. A user registers on kirafinancial.com to log in to Plaid, and thereby their bank(s), and to provide Kira with their phone number. Then Kira sends an initial text message introducing herself. Thereafter, whenever the user sends Kira a text message, Twilio receives it and passes it off to a server running on kirafinancial.com. This server sends the user's message to ChatGPT, with an appropriate guiding prompt, and examines ChatGPT's response. Depending on the response, more interactions with ChatGPT may occur, Plaid data may be retrieved, and Plaid actions may be taken. Finally, a response (generated by ChatGPT, typically) is sent back to the user via text message (Twilio).

3.1 ChatGPT

ChatGPT acts as the glue between the user's request and their financial data. We define a prompt that contains a brief introduction to various account types that the Plaid API might hold, and instructions for categorizing the user's query. The prompt helps guide the model's response generation, ensuring that the output is relevant to the user's query.

When a user inputs a query, we send a set of messages to the ChatGPT API, including the predefined prompt, the role of Kira as a financial advisor, and the user's query. We then receive a response from the API which categorizes the query. Based on the category, we continue to handle the query accordingly. If the query is about loans, we send another set of messages to ChatGPT to generate a short, concise answer to the user's question. The messages include guidelines for providing the response, such as limiting it to a maximum of five sentences and including essential information about loans like interest rates, fees, and charges.

Figure 4 shows our initial prompt and the API call to ChatGPT. One can see how we provide ChatGPT a long list of options for how to categorize the user's input.

If ChatGPT tells us the user's query is about accounts, we further analyze the response to identify the subcategory, such as account fees, account numbers, account balances, or displaying all accounts. Depending on the subcategory, we generate specific responses by sending another set of messages to ChatGPT, similar to what we did with the loans. But in these messages, we also provide a table of account data from Plaid to help the model generate accurate and contextually correct responses. For example, if the user's query is about account fees, we send the table of account information and instruct ChatGPT to give a short and polite answer that only addresses the fees related to the specific account in question. This process happens for every subcategory under accounts, the only thing that varies between them is the messages that ChatGPT receives for each one. These messages must be customized to each subcategory so that Kira can fully complete the task in question.

Figure 5 shows an example of a prompt to ChatGPT that's handling the case about fees.

If, on the other hand, the category of the query is undefined, ChatGPT will not be able to answer the user's question. In this case, Kira will return a message asking the user to ask a more specific question related to accounts or loans in general. This ensures that the user receives a helpful response and is guided to ask a more relevant question within the functionalities of Kira. We do so to ensure that ChatGPT doesn't answer random questions and so that Kira can only be used as a financial assistant, not a general-purpose chatbot.

This process continues in a loop, permitting the user to ask multiple ques-

```
prompt = """
These are account types that you should know:
- savings
- credit card
- loan
- investment
- mortgage
- brokerage
- money market
- prepaid card
- other
Given the query below, go through these steps and answer as such
1. Is it about Accounts, Loans or a thank you from the user?
- If the user is saying thank you, mark that.
- If it falls under loans, answer the question shortly unless it’s about
    the account balance of a certain loan like for example a student loan
    then it fall under the accounts category.
- If it falls under ’Accounts’ put it in one of these subcategories,
which needs to be one of these:
a. Account Fees
b. Get account number
c. Account Balance
    - If it fall under a,b,c tell me which account type was mention in the
query.
d. Display all accounts
e. Move money between accounts
    - If it falls under this subcategory tell me which account types were
mentioned in the query. (just answer this part with account types)

If there is no query, say "No query." If you do not know the category, say
"Unknown category."
"""
model = "gpt-3.5-turbo"
completions = openai.ChatCompletion.create(
    model=model,
    messages= [
        {"role": "system","content": "Your name is Kira and you are a
financial advisor."},
        {"role": "assistant","content": prompt},
        {"role": "user","content": "Query: %s"%query}
    ],
    max_tokens=1024,
    n=1,
    stop=None,
    temperature=0.0,
)
```

Figure 4: Initial prompt to ChatGPT.

```
if "fees" in message.lower():
  # The query is about account fees
  # Answer the query
  messages2 =  [
      {"role": "system","content": "Your name is Kira and you are a
financial advisor. Always refer to the table of data provided. Don’t say
you’re sorry or that you don’t have access to account data."},
      {"role": "assistant","content": "Answer the user’s query
according to the following table of account information. Make sure that you
only answer with details about the fees on this account and nothing else."},
      {"role": "assistant","content": "Keep your answer very short and
polite. If there is a number in your answer put it in dollars ($)"},
      {"role": "assistant","content": account_table},
      {"role": "user","content": "Query: %s"%query}
  ]
```

Figure 5: Example of a ChatGPT prompt for handling the case when the user asks about fees.

tions, while ChatGPT provides relevant and specific answers based on the given instructions and data. The loop ensures a seamless conversational experience between the user and Kira.

3.2 Plaid

In our pursuit of developing Kira, we aimed to create a tool that could help users with their financial tasks and answer questions about their finances. To achieve this, we required a method to securely and efficiently access users’ financial data from various institutions. Plaid, a well-established fintech platform, provided us with the solution through its API, which we implemented using Plaid Quickstart [8] and integrated with our React-based website described in Section 5.

To begin the Plaid implementation, we first set up a Plaid account to obtain the necessary API keys. Next, we added the Plaid Link component to our React website, enabling a secure connection between users’ financial accounts and our application. Users provide their financial institution login credentials to Plaid, which generated a unique access token, granting our application secure access to their financial data.

Once we received the access token, we execute Plaid API calls to retrieve financial data in JSON format, including account balances, transaction histories, and other essential financial information. We designed Kira to request real-time data from users’ financial accounts during each interaction, ensuring

```
{"accounts": [
      {
      "account_id": "BxBXxLj1m4HMXBm9WZZmCWVbPjX16EHwv99vp",
      "balances": {
          "available": 100,
          "current": 110,
          "iso_currency_code": "USD",
          "limit": null,
          "unofficial_currency_code": null
      },
      "mask": "0000",
      "name": "Plaid Checking",
      "official_name": "Plaid Gold Standard 0% Interest
          Checking",
      "persistent_account_id":
"8cfb8beb89b774ee43b090625f0d61d0814322b43bff984eaf60386e",
      "subtype": "checking",
      "type": "depository",
      "account_number": "1111222233334444"}]}
```

Figure 6: Example JSON data from Plaid API.

the information provided by our financial assistant is accurate and up to date. Figure 6 shows a short example of how the Plaid API JSON file might look when initially provided to Kira.

After obtaining this kind of financial data, we parse and process the data, making it more suitable for the ChatGPT API to effectively utilize. We iterate through each account, extracting information such as account name, available balance, current balance, account type, account subtype, account number, and associated fees. ChatGPT can understand many text formats, but we found a CSV format showing the financial data in tabular form worked well.

3.3 Twilio

To implement Twilio with Kira, we first created an account on Twilio's console and purchased a phone number associated with Kira. Since Kira will mainly be dealing with clients based in the US, we purchased an American phone number. If needed, other national numbers can be easily bought and connected to Kira for communication with people outside of the states.

Whenever Kira needs to send a text message, we just execute a Twilio API call. However, we cannot predict when the user will send a message to Kira, so Twilio executes a 'hook' on our server (kirafinancial.com) whenever a message arrives. Thus, Kira waits for messages to arrive, and reacts to them by consulting with ChatGPT and then sending a response message.

4 Obfuscation and Deobfuscation

Obfuscation in general is a technique used to intentionally obscure or conceal the meaning or purpose of data. It is typically used as a security measure to make it more difficult for unauthorized individuals to understand or reverse engineer software or data [1].

In the context of programming, obfuscation involves transforming code into an equivalent but harder-to-understand form, while preserving its functionality. This can include techniques such as renaming variables and functions with meaningless or misleading names, adding unnecessary complexity or redundancy to the code, removing comments or whitespace, and using code-generating tools to create convoluted or oblique code structures. The goal is to make the code difficult to understand, interpret, or modify, thereby increasing the effort and time required for reverse engineering or unauthorized access.

In the field of information security, obfuscation can also be used to protect sensitive data by disguising it or encrypting it in a way that is not easily recognizable or understandable without proper decryption keys or algorithms. This can include techniques such as data encoding, encryption, and steganography, where data is hidden within other seemingly innocuous data or files [1].

On the contrary, deobfuscation is the process of reversing or undoing obfuscation techniques applied to code or data in order to restore their original form or meaning [17]. It involves analyzing and deciphering obfuscated code or data to understand its functionality or extract meaningful information.

We utilized obfuscation to add a privacy aspect to Kira. The data we received from Plaid is obfuscated using 1 of 4 different obfuscation functions, each one of which deals with a different data type: Integer, Currency, Date, and String. Each function stores the tokenized data and the original data in a token map that is used in the deobfuscation function. The new obfuscated data is sent to ChatGPT to ensure user privacy on sensitive information. The response from ChatGPT is sent through a deobfuscation function to retrieve the original data sent by the user. This process makes it so ChatGPT will never interact with user-sensitive information.

The integer obfuscation function, tokenize_acctNum(num) is a function that takes a numeric value (num) and generates a random 9-digit token using the random.choices() function from the random module. It then creates a dictionary (token_map) with the generated token as the key and the original numeric value as the value. The function returns the generated token and the token map [4].

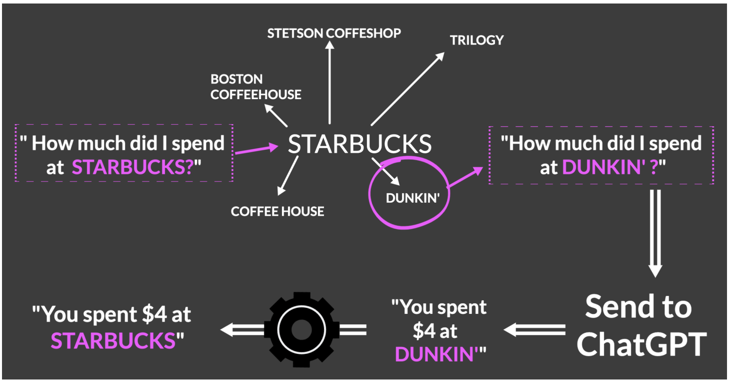

STETSON COFFESHOP
TRILOGY
BOSTON COFFEEHOUSE
" How much did I spend at STARBUCKS?"
STARBUCKS
"How much did I spend at DUNKIN' ?"
DUNKIN'
COFFEE HOUSE
Send to ChatGPT
"You spent $4 at DUNKIN'"
"You spent $4 at STARBUCKS"

Figure 7: Obfuscation - Flowchart

The currency obfuscation function, tokenize_currency(dollars), is a function that takes a currency value in dollars (dollars) as input. It removes any dollar sign or decimal point from the input value using the replace() method, and then converts the resulting value to an integer. Next, it generates a random dollar amount between 0 and 999 using the random.randint() function, and formats it with leading zeros to have three digits. It replaces the cents portion of the input value with the generated random dollar amount to create a new token. The function creates a dictionary (tokenCurrMap) with the new token as the key and the original currency value as the value, and returns the new token and the token map [4].

The date obfuscation function, tokenize_date(date_str), takes a date string (date_str) in the format 'YYYY-MM-DD' as input. It converts the input string to a datetime object using datetime.strptime() from the datetime module. It then generates a random delta (time interval) between 0 and 365 days using the timedelta() function from the datetime module and the random.randint() function. It adds the random delta to the input date to obtain a new random date. The new date is then formatted as a string in the 'YYYY-MM-DD' format, and this string is used as the token. The function creates a dictionary (token_map_date) with the new date token as the key and the original date string as the value, and returns the new date token and the token map [4].

The last obfuscation function is the string obfuscation, tokenize_name(). This function uses the SentenceTransformer library to create a sentence embedding model called embedder using the ‘all-MiniLM-L6-v2’ pre-trained model [5]. It then defines a list of sentences (corpus) containing various names of places, restaurants, bookstores, clubs, and fashion brands. The function randomly selects a sentence from the corpus list using the random.choice()

```
corpus_embeddings = embedder.encode(corpus, convert_to_tensor=True)
top_k = min(10, len(corpus))
for query in queries:
    query_embedding = embedder.encode(query, convert_to_tensor=True)
    cos_scores = util.cos_sim(query_embedding, corpus_embeddings)[0]
    top_results = torch.topk(cos_scores, k=top_k)
```

Figure 8: Python code to find similar vendor names for the obfuscation procedure.

function and generates a sentence embedding for the selected sentence using the embedder.encode() function. The resulting sentence embedding is used as the token, and the original sentence is used as the value in a dictionary (token_map_name). The function returns the generated token and the token map.

To convert the obfuscated data back to its original value we wrote a deobfuscation function called detokenize. The detokenize function takes two arguments as input: token, which is a tokenized string or word, and token_map, which is a dictionary that maps tokens to their corresponding original words or phrases. The function then uses the get() method of the token_map dictionary to retrieve the value associated with the token key, which represents the original word or phrase corresponding to the token [17]. If the token key is not present in the token_map dictionary, the get() method returns None. The function returns this retrieved value or None as the detokenized version of the input token. This deobfuscated data is then sent back to the user. The chart below illustrates how the SentenceTranformer groups similar categories togther from our corups.

5 Website: KiraFinancial.com

In this project, we utilized React, a popular JavaScript library, to develop a user-friendly website called KiraFinancial.com. React allowed us to create reusable UI components and manage the application state effectively.

The primary components in our application are the RegistrationPage and the SuccessPage. The RegistrationPage serves as the main landing page, where users can provide their personal information to connect their phone number and Plaid account with Kira, our financial assistant.

The RegistrationPage component is responsible for managing the state of user inputs, such as first name, last name, country code, and phone number.

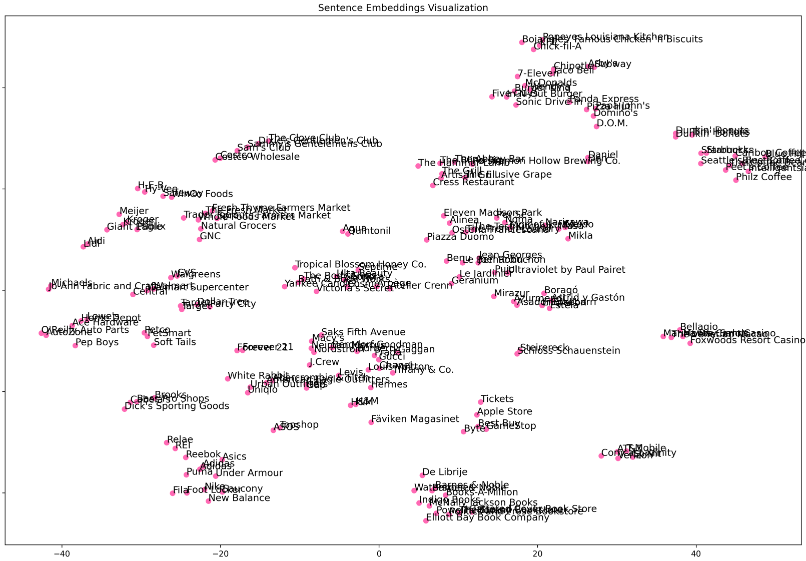


Figure 9: Obfuscation - Visual Embeddings

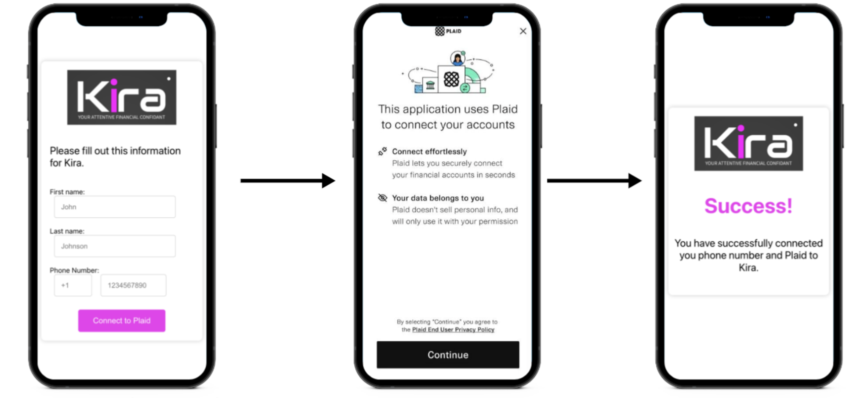
It also tracks the connection status of the user's Plaid account. To collect user information, we use controlled input components that update the component's state when users interact with the input fields.

Upon submitting the form, the component validates the user's input, ensuring that all fields are filled out and properly formatted. If any validation errors are found, an alert is displayed with a corresponding error message.

After successful validation, the user's information is sent to the backend server using a POST request. The component then initiates the Plaid Quickstart process, guiding the user through their account login/signup with Plaid. When the API call is successful, and the user's account is connected, the component updates its state to indicate a successful connection.

Depending on the connection status, the component either renders the SuccessPage or displays the registration form. The SuccessPage component simply shows a success message to the user, confirming that their phone number and Plaid account have been successfully connected to Kira. This streamlined process ensures a smooth user experience while connecting with Kira.

We have implemented a section of backend code that is responsible for receiving user input from our website using Flask, which is the same Flask application used to implement Twilio just with a different route. We create an app object using Flask, which defines a route /register with the HTTP method



Please fill out this information for Kira.

First name: John

Last name: Johnson

Phone Number: +1 1234567890

This application uses Plaid to connect your accounts

Connect effortlessly
Plaid lets you securely connect your financial accounts in seconds

Your data belongs to you
Plaid doesn't sell personal info, and will only use it with your permission

By selecting "Continue" you agree to the Plaid End User Privacy Policy

Success!

You have successfully connected you phone number and Plaid to Kira.

Register your phone number

Connect to Plaid

Ready to text Kira!

Figure 10: Website - KiraFinancial.com

POST to handle incoming POST requests to the /register endpoint.

Inside the register() function, we use request.json, a method provided by Flask, to parse the JSON data received from the front-end. We then extract the relevant data such as firstName, lastName, phoneNumber, and token from the received data using data.get() method, and store them in respective variables.

Next, we update a dictionary named registration_data with the phoneNumber as the key and the token as the value. To persist the registration_data dictionary, we use pickle.dump() function to serialize it, and save the serialized data to a binary file named "reg.pkl" using a with statement to ensure proper file handling [12].

Our implemented backend code receives user input from our website via a Flask route, extracts relevant data from the received JSON data, updates a dictionary with the extracted data, and then serializes and saves the dictionary to a file named "reg.pkl".

The pickle file in Kira serves as a repository for phone numbers and token data, which are crucial in various functions. The phone number stored in the file is utilized in the SMS function [15], serving as the register receiving number, enabling seamless text communication between Kira and the user. Additionally, the token, also stored in the file, grants access to the user's sensitive Plaid information, such as banking data. The use of a python pickle file allows us to effectively store and update all user information without losing any prior data. When the Flask application is initiated, the pickle file loads all user data, and when a new user registers via the website, the information is appended to the same pickle file [12].

6 Interactions with Kira

To evaluate Kira's performance, we conducted a series of tests to measure the accuracy and usefulness of her responses to common financial questions. We asked Kira four different questions, each ten times, and rated her answers as Perfect, Okay, or Bad. The questions included: "What is my credit card balance?"(1), "What are the fees on my credit card?"(2), "Display all my accounts"(3), and "Tell me about student loans"(4). We selected these questions because they are common queries that individuals may have about their finances.

Overall, Kira performed well in our tests, with the majority of her responses rated as Perfect or Okay. Our stacked bar-chart shows the breakdown of Kira's responses for each question, with the percentage of Perfect, Okay, and Bad ratings. We found that Kira was particularly effective at providing accurate and up-to-date information about account balances and fees. However, we also noted that Kira's responses to more complex questions, such as the question about student loans, were not always as detailed or informative as we would have liked.

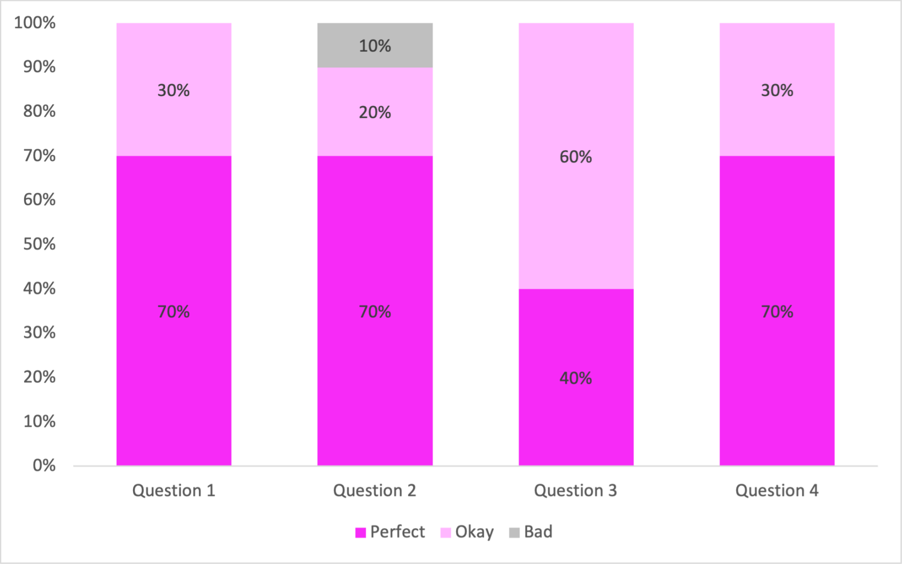


Figure 11: Interactions performance

The script below is a live conversation that we had with Kira with her exact responses.

USER: Hey Kira, can you display all my accounts?

KIRA: Certainly, here are the account names: Plaid Checking, Plaid Credit Card, and Plaid Student Loan.

USER: What is my credit card balance?

KIRA: Your current balance on your credit card is $410.

USER: What are the fees on my credit card?

KIRA: The fees on your Plaid Credit Card are a Monthly Maintenance Fee of $10.0, dated 2022-02-01 and an Overdraft Fee of $35.0, dated 2022-02-14.

USER: Do I have an student loans and what is my balance on those loans?

KIRA: You have a student loan, and your current balance is $65,262.

USER: Tell me about student loans

KIRA: Student loans are a type of loan designed to help students pay for their education. They can cover tuition fees, books, and living expenses. Fees and charges vary depending on the lender and the type of loan. Interest rates can also vary, but they are typically lower than other types of loans. It's important to research and compare different options before applying for a student loan.

USER: Do you know who Telma is?

KIRA: I'm sorry, I didn't understand your question. Please ask me about your accounts or loans in general

USER: Thank you for all your help Kira!

KIRA: You're welcome, have a great day! Please let me know if there is anything else I can do for you today.

7 Conclusion

Kira demonstrates the potential of integrating Artificial Intelligence (AI) technologies, such as OpenAI's ChatGPT, the Twilio platform, and the Plaid platform, to create an advanced financial assistant named Kira. Kira operates exclusively through text messages and offers a wide range of functionalities, including answering questions, providing assistance with loans, budgets, account information, expenses, and account money transfers.

One of the key contributions of this research is the development of an algorithm that obfuscates the user's data before sending it to ChatGPT, addressing concerns about user privacy and data security. This approach ensures that Kira can understand and respond to complex natural language questions and commands while preserving the user's financial privacy.

Furthermore, this work highlights the importance of accessibility and convenience in banking and finance, as Kira eliminates the need for repeated logins and authentication, providing users with a seamless and efficient experience.

The integration of multiple technologies and platforms in building Kira showcases the potential of AI-powered chatbots and virtual assistants in transforming the way financial services are delivered to users.

However, it is important to acknowledge the limitations of this work, such as the need for further evaluation of the obfuscation algorithm's effectiveness and potential vulnerabilities, as well as potential ethical considerations in using AI in financial services. Future research could focus on addressing these limitations and exploring additional functionalities and improvements to enhance Kira's performance and user experience.

References

[1] David E Bakken et al. "Data obfuscation: Anonymity and desensitization of usable data sets". In: *IEEE Security & Privacy* 2.6 (2004), pp. 34–41.

[2] Brendan Choi. "Python Network Automation Labs: Ansible, pyATS, Docker, and the Twilio API". In: *Introduction to Python Network Automation: The First Journey.* Springer, 2021, pp. 675–732.

[3] *Erica - Virtual Financial Assistant From Bank of America.* en-US. n.d. URL: `https://promotions.bankofamerica.com/digitalbanking/mobilebanking/erica`.

[4] Geekculture. *Python Source Code Obfuscation.* n.d. URL: `https://medium.com/geekculture/python-source-code-obfuscation-6b97f88a460d` (visited on 04/16/2023).

[5] Md Tahmid Rahman Laskar, Xiangji Huang, and Enamul Hoque. "Contextualized embeddings based transformer encoder for sentence similarity modeling in answer selection task". In: *Proceedings of the Twelfth Language Resources and Evaluation Conference.* 2020, pp. 5505–5514.

[6] OpenAI. *Introducing ChatGPT.* 2022. URL: `https://openai.com/blog/chatgpt`.

[7] OpenAI. *Pricing.* `https://openai.com/pricing`. [Accessed 1-Apr-2023]. n.d.

[8] Plaid. n.d. URL: `https://plaid.com/docs/quickstart/` (visited on 03/25/2023).

[9] Plaid. *Plaid API Documentation.* 2021. URL: `https://plaid.com/docs/` (visited on 04/03/2023).

[10] Plaid. *Plaid: Global Coverage.* n.d. URL: `https://plaid.com/global/` (visited on 04/01/2023).

[11] Plaid. *Plaid: Unlock financial freedom for everyone.* n.d. URL: https://plaid.com/ (visited on 04/01/2023).

[12] Python Software Foundation. *pickle — Python Object Serialization.* n.d. URL: https://docs.python.org/3/library/pickle.html (visited on 02/16/2023).

[13] Rasa. *Rasa Platform.* n.d. URL: https://rasa.com/.

[14] *The inside story of how ChatGPT was built from the people who made it.* en. 2023. URL: https://www.technologyreview.com/2023/03/03/1069311/inside-story-oral-history-how-chatgpt-built-openai/.

[15] Twilio. *How to Send SMS Messages with Twilio in Python.* n.d. URL: https://www.twilio.com/docs/sms/tutorials/how-to-send-sms-messages-python (visited on 03/18/2023).

[16] Twilio. *Twilio.* n.d. URL: https://www.twilio.com/.

[17] Sharath K Udupa, Saumya K Debray, and Matias Madou. "Deobfuscation: Reverse engineering obfuscated code". In: *12th Working Conference on Reverse Engineering (WCRE'05).* IEEE. 2005, 10–pp.

An LOD-based AIG Approach to Automatically Generating Object-Oriented Programming Problems*

Laura Zavala[1], Benito Mendoza[2]
[1]Medgar Evers College
Brooklyn, NY 11225
zavalagu@mec.science
[2]New York City College of Technology
Brooklyn, NY 11201
bmendoza@citytech.cuny.edu

Abstract

Success in introductory programming courses requires a significant amount of practice. Unfortunately, creating practice materials (e.g., assignments and quizzes) takes a lot of time and effort. We use Automatic Item Generation (AIG) and Linked Open Data (LOD) to automatically generate numerous object-oriented programming exercises. AIG consists of automatically creating test-items or questions by using templates with embedded variables and formulas, which are resolved by a computer program with actual values to generate questions. We use LOD to populate the templates, thus providing a vast set of values for the variables in the templates. Besides being able to generate a large number of exercises from a single template, the resulting exercises can be focused on specific topics or contexts, and therefore be more meaningful to students. We present our LOD-based AIG approach and show the results of a small-scale study conducted to evaluate the likeness of the exercises generated by our approach compared to that of exercises found in textbooks.

1 Introduction

Basic object-oriented programming concepts, such as classes, properties, constructors, and instantiation, are among the fundamental building blocks that students must learn but that are difficult for them to grasp. As Holland, Griffiths and Woodman [9] claim, misconceptions of object concepts can be hard to shift later and can hinder future teaching on the subject. Ragonis and Ben-Ari [14] studied students in a first object-oriented programming course and found that one of the major difficulties for students to understand is object instantiation by its constructor. Eckerdal et al. [5] identified object-orientation as Threshold Concepts in the computer science domain, which in Theory of Learning are core concepts whose characteristics can make them troublesome in learning.

As validated by many works [1, 8, 11, 16, 17, 20, 22], proficiency in programming courses is usually reached through extensive practice. However, creating practice materials is laborious and time-consuming. We use Automatic Item Generation (AIG) to automatically generate object-oriented programming practice exercises, solved and unsolved. We have focused on creating exercises with basic object-oriented concepts: classes with a few properties and methods, constructors, instantiation, and objects. AIG uses templates with embedded variables and formulas that are resolved by a computer program with actual values to generate questions. We have extended the traditional template-based AIG approach with the use of Linked Open Data (LOD) sources to populate the templates, providing a vast set of values for the variables in the templates. LOD results from sharing and interlinking data on the Web so that it can be queried and used by computer programs. LOD has emerged as one of the largest collections of interlinked datasets on the web, with hundreds of very large datasets on different topics/domains openly available for querying. Besides generating numerous exercises, the use of LOD allows us to create exercises that are somewhat meaningful and from a variety of topics (e.g., sports, movies, music, cars, politicians, etc.). Several educational theories emphasize the need for introductory contexts that align with students' interests and goals [3, 15]. Examples in introductory courses should make sense to students and promote engagement. Results show that these approaches increase student motivation, facilitate understanding, and improve outcomes and retention rates.

This paper presents our LOD-based AIG approach to the generation of object-oriented programming exercises. The approach facilitates the generation of a large pool of practice exercises on a variety of topics. Further, students can be given the choice of topic that they would like their questions to be about. We discuss an initial evaluation, where we allowed students to express a preference between exercises generated with our approach and exercises found in textbooks.

2 Related Work

Some works have explored the use of AIG in the computer-programming domain. The work of Hsiao et al [10] describes QuizJET, a templating system for object-oriented programming. QuizJET was designed to focus on a narrow subset of programming concepts and used 100 handcrafted templates for generating questions. In [13] AIG is used to automatically generate questions in the mathematics, physics, and computer programming domains. The only variability in the computer programming questions is the programming language asked to solve the problem (the assumption is that students can write in different programming languages). The authors point out that the main point of interest of these exercises is in its automatic grading through test cases.

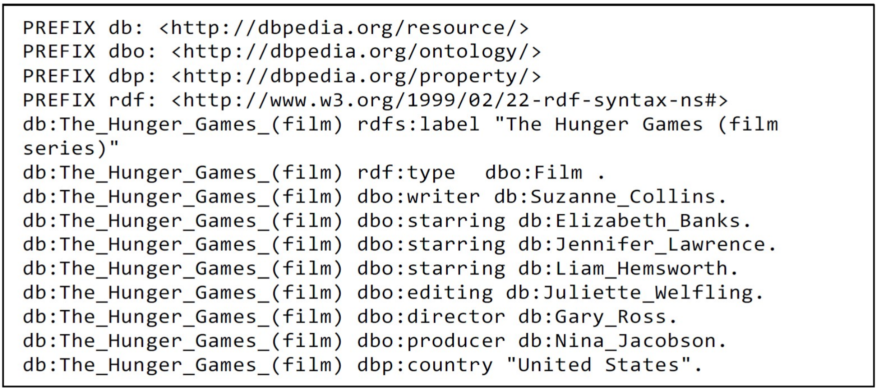
Ontologies have been used for multiple choice question generation along with some natural language processing to generate factual questions about the domain of the ontology [4, 12, 18, 19]. For example, a geographic ontology is used in [6] to generate questions about geography. No AIG templates are used. In contrast, we use Linked Open Data and its associated ontologies to insert context into the questions we generate using AIG templates.

The use of Linked Open Data for question generation has been explored in [18]. Their focus is on the use of LOD as the domain knowledge from which questions can be generated. They raise the issue of data quality and inconsistencies in LOD which can be a problem when LOD is used as the source of knowledge. In contrast, we use LOD to contextualize the questions, which do not belong to the same domain as the ontology (the domain is computer programming). For example, using a movies ontology, [6] would generate quizzes about movies while we instead generate computer programming questions in the context of movies (using movies as part of the problem formulation).

We have previously introduced the use of semantic-based AIG for the automatic generation of basic contextual exercises for introductory programming concepts, such as conditional, loops, and arrays/lists [21]. The approach presented in this paper is focused on the generation of OOP problems and uses other features of the LOD datasets and ontologies, such as concepts and their attributes, which naturally align with OOP concepts, i.e., classes and their properties.

3 LOD-based Automatic Item Generation

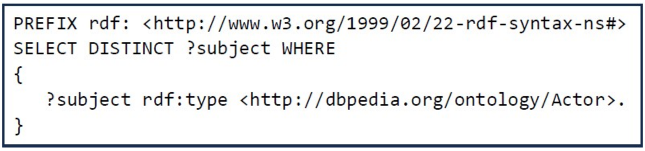
Linked Open Data is machine-readable data published on the Web using recognized standards so that it can be interlinked and become more useful through semantic queries [2]. The collection of Linked Data published on the Web, contains around 1,300 datasets, which cover several domains, including entertainment (e.g., music, celebrities, movies, books, sports, and events), governments, life sciences, health, news, and geographic and geospatial data. Some of the largest and most used LOD datasets include DBpedia, GeoNames, and Wikidata. In LOD, information is represented using the Resource Description Framework (RDF). In RDF, (subject, predicate, object) triples are used to describe resources. The subject of a triple is the URI (Uniform Resource Identifier) identifying the described resource. The object can either be a simple value or the URI of another resource that is related to the subject. The predicate specifies how the subject and object are related. As an example, Figure 1 shows an excerpt from DBpedia about The Hunger Games movie. Optionally, resources in LOD can be associated with an ontology that specifies the concepts (classes) that a resource can belong to, as well as the types of relations among classes. In Figure 1, The Hunger Games is specified to belong to the Film class. Further, the DBpedia ontology states that a resource of the class Film (e.g., The Hunger Games) has a writer relation to a resource of the class Person (e.g., Suzanne Collins). LOD datasets can be queried using the SPARQL query language to query local or remote repositories (e.g., http://dbpedia.org/sparql/). Figure 2 shows an SPARQL query that can be used to get a list of actors from the DBpedia dataset.



```
PREFIX db: <http://dbpedia.org/resource/>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbp: <http://dbpedia.org/property/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
db:The_Hunger_Games_(film) rdfs:label "The Hunger Games (film
series)"
db:The_Hunger_Games_(film) rdf:type  dbo:Film .
db:The_Hunger_Games_(film) dbo:writer db:Suzanne_Collins.
db:The_Hunger_Games_(film) dbo:starring db:Elizabeth_Banks.
db:The_Hunger_Games_(film) dbo:starring db:Jennifer_Lawrence.
db:The_Hunger_Games_(film) dbo:starring db:Liam_Hemsworth.
db:The_Hunger_Games_(film) dbo:editing db:Juliette_Welfling.
db:The_Hunger_Games_(film) dbo:director db:Gary_Ross.
db:The_Hunger_Games_(film) dbo:producer db:Nina_Jacobson.
db:The_Hunger_Games_(film) dbp:country "United States".
```

Figure 1: An excerpt from DBpedia about The Hunger Games movie

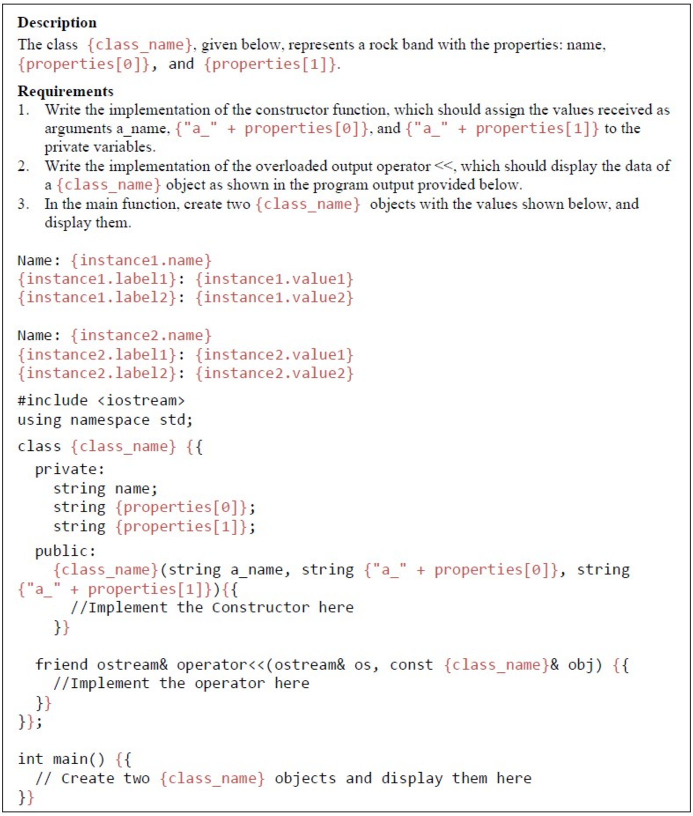
AIG is an approach for developing test-items automatically by a program [7]. Existing approaches to AIG are mainly template-based. Instead of creating a question (test-item), experts create a test-item template with embedded variables and formulas. By replacing those variables and formulas with different values from a range of values specified by the expert, a high volume of test-items can be generated from a single item template. Figure 3 shows a template for generating C++ object-oriented programming problems using our



```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT DISTINCT ?subject WHERE
{
   ?subject rdf:type <http://dbpedia.org/ontology/Actor>.
}
```

Figure 2: An SPARQL query to obtain a list of actors from DBpedia

approach. Our LOD-based AIG approach uses Wikidata as the data source for populating the question templates. Table 1 provides sample SPARQL queries that can be used to query Wikidata to obtain values to populate the variables in the template. Figure 4 shows an exercise generated using the template in Figure 3 and one of the queries in Table 1.



Description

The class {class_name}, given below, represents a rock band with the properties: name, {properties[0]}, and {properties[1]}.

Requirements

1. Write the implementation of the constructor function, which should assign the values received as arguments a_name, {"a_" + properties[0]}, and {"a_" + properties[1]} to the private variables.
2. Write the implementation of the overloaded output operator <<, which should display the data of a {class_name} object as shown in the program output provided below.
3. In the main function, create two {class_name} objects with the values shown below, and display them.

```
Name: {instance1.name}
{instance1.label1}: {instance1.value1}
{instance1.label2}: {instance1.value2}

Name: {instance2.name}
{instance2.label1}: {instance2.value1}
{instance2.label2}: {instance2.value2}
```

```
#include <iostream>
using namespace std;

class {class_name} {{
  private:
    string name;
    string {properties[0]};
    string {properties[1]};
  public:
    {class_name}(string a_name, string {"a_" + properties[0]}, string
{"a_" + properties[1]}){{
      //Implement the Constructor here
    }}

  friend ostream& operator<<(ostream& os, const {class_name}& obj) {{
    //Implement the operator here
  }}
}};

int main() {{
  // Create two {class_name} objects and display them here
}}
```

Figure 3: AIG template that generates questions like the one in Figure 4

While most LOD datasets start with RDF as a base data model, Wikidata developed its own data model, which, at the simplest level well-matched to RDF, but it also provides better means for capturing more complex relations such as n-ary relations. RDF triple representations work straightforwardly for simple relations involving two entities; however, many interesting facts involve more than just two entities and a relation between them. For example, for an actor/actress starring in a movie (three such cases are shown in Figure 1: Elizabeth Banks, Jennifer Lawrence, and Liam Hemsworth), the name of the character they portray may be relevant. For our application, namely classes and objects in object-oriented programming, this might come in handy.

Currently, we have a fixed set of classes/concepts from Wikidata that can be used to instantiate the variables in a template, and we have pre-built SPARQL queries for them. Some examples of the classes in our domain are: NBATeam, NFLTeam, MBLTeam, MLSTeam, PremierTeam, RockBand, PopBand, PopStar, LuxuryCar, and SuperCar. Table 1 shows SPARQL queries for some of these. We also use instances of those classes and the values for their properties to include object instantiation in the exercises. Not all instances will be useful since they might have missing values for the properties. Similarly, not all properties are relevant to the class and instance in use, as they might be generic and common to many (or all) classes, for example, description and image. We have identified and compiled a list of irrelevant properties for each class so that they are not used. We select the instances and properties to be used, using Procedure 1 and Procedure 2 as explained below.

Procedure 1

```
1. Select a class $A$ from our Domain
2. Run the corresponding query to obtain the list of instances $I$ of class $A$
3. Select two instances, $i_1$, and $i_2$, from $I$, and a set of suitable properties,
   $P$, using Procedure 2
```

Procedure 2

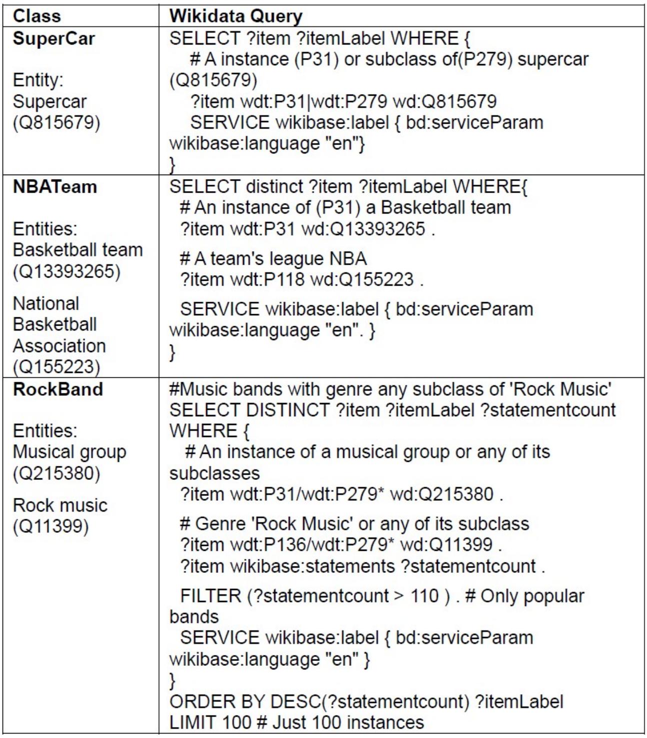
```
1. $P = [\,]$
2. while the length of $P < 3$:
3.     Randomly pick two instances $i_1$, $i_2$ from $I$
4.     Get the list of properties $P_1$ of instance $i_1$
5.     Get the list of properties $P_2$ of instance $i_2$
6.     Get the list of properties available in both instances $P = (P_1 \cap P_2)$
7.     Remove from $P$ the irrelevant properties
8.     Remove from $P$ the properties that have the same value for $i_1$ and $i_2$
9. Return $i_1$, $i_2$, $P$
```

Following these procedures, we select two instances of a given class A. These instances must have at least three properties in common. However, those properties should not have the same value for both selected instances. For example, for the NBATeam class, all the instances have a value of “NBA” for the property league, so the property is not selected.

4 Evaluation and Discussion

Besides having the potential to generate a vast number of questions or exercises from a single template, the exercises generated using our LOD-based AIG approach are more meaningful and can focus on specific topics or contexts (e.g., sports, movies, music, places, etc.).

Table 1: Sample of Wikidata queries for some of the classes in the domain.



Class	**Wikidata Query**

SuperCar
Entity:
Supercar
(Q815679)
	SELECT ?item ?itemLabel WHERE {
	# A instance (P31) or subclass of(P279) supercar (Q815679)
	?item wdt:P31|wdt:P279 wd:Q815679
	SERVICE wikibase:label { bd:serviceParam wikibase:language "en"}
	}

NBATeam
Entities:
Basketball team (Q13393265)
National Basketball Association (Q155223)
	SELECT distinct ?item ?itemLabel WHERE{
	# An instance of (P31) a Basketball team
	?item wdt:P31 wd:Q13393265 .
	# A team's league NBA
	?item wdt:P118 wd:Q155223 .
	SERVICE wikibase:label { bd:serviceParam wikibase:language "en". }
	}

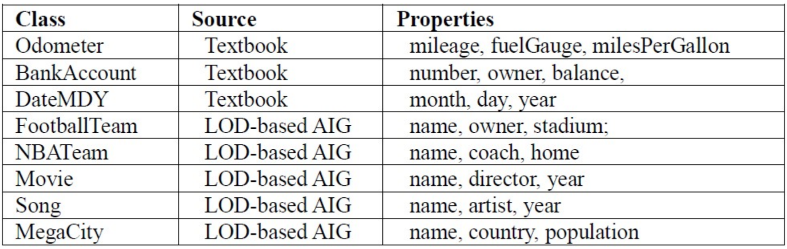
RockBand
Entities:
Musical group (Q215380)
Rock music (Q11399)
	#Music bands with genre any subclass of 'Rock Music'
	SELECT DISTINCT ?item ?itemLabel ?statementcount WHERE {
	# An instance of a musical group or any of its subclasses
	?item wdt:P31/wdt:P279* wd:Q215380 .
	# Genre 'Rock Music' or any of its subclass
	?item wdt:P136/wdt:P279* wd:Q11399 .
	?item wikibase:statements ?statementcount .
	FILTER (?statementcount > 110) . # Only popular bands
	SERVICE wikibase:label { bd:serviceParam wikibase:language "en" }
	}
	ORDER BY DESC(?statementcount) ?itemLabel
	LIMIT 100 # Just 100 instances

We believe that this is an advantage over some of the traditional object-oriented programming examples, which tend to be more abstract or about concepts that only few students are familiar with. Using our approach, students could even be given the choice of topic that they would like their questions to be about.

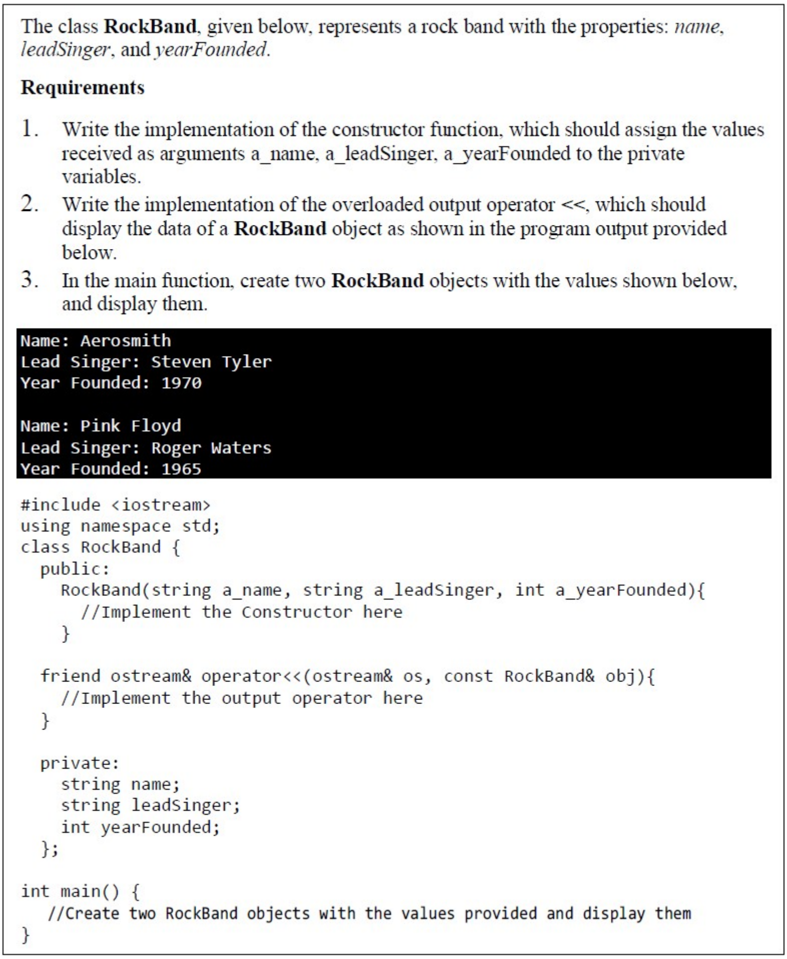
We conducted a small-scale study to evaluate the likeness of the exercises generated by our approach compared to that of exercises found in textbooks. Fifteen students in a Data Structures class were given a quiz about basic object-oriented programming concepts in C++. The quiz consisted of only one question with the skeleton of a class with three properties. Students were asked to write the constructor for the class, the overloading output operator, and a main function where two objects of the class were created with specific given values. They were given the option to choose the quiz they wanted to take, among two options: one generated with our approach and one taken from existing books or tutorials. We generated six different problems with our approach and collected three problems from existing sources, so that we could create different combinations to give to each student in the class. Figure 4 shows one of the exercises created with our approach. Table 2 lists all the classes for the exercises used in the study.

All students in the class chose to work on the problem that was generated using our LOD-based AIG approach. This preliminary study shows students prefer certain familiar topics over others less familiar (not necessarily that they prefer our generated examples over existing ones). We plan to conduct more studies and in the long term we are interested in studying the effects that this can have in the learning process.

Table 2: Pool of classes for the quiz given to students.



Class	**Source**	**Properties**
Odometer	Textbook	mileage, fuelGauge, milesPerGallon
BankAccount	Textbook	number, owner, balance,
DateMDY	Textbook	month, day, year
FootballTeam	LOD-based AIG	name, owner, stadium;
NBATeam	LOD-based AIG	name, coach, home
Movie	LOD-based AIG	name, director, year
Song	LOD-based AIG	name, artist, year
MegaCity	LOD-based AIG	name, country, population



The class **RockBand**, given below, represents a rock band with the properties: *name*, *leadSinger*, and *yearFounded*.

Requirements

1. Write the implementation of the constructor function, which should assign the values received as arguments a_name, a_leadSinger, a_yearFounded to the private variables.
2. Write the implementation of the overloaded output operator <<, which should display the data of a **RockBand** object as shown in the program output provided below.
3. In the main function, create two **RockBand** objects with the values shown below, and display them.

```
Name: Aerosmith
Lead Singer: Steven Tyler
Year Founded: 1970

Name: Pink Floyd
Lead Singer: Roger Waters
Year Founded: 1965
```

```
#include <iostream>
using namespace std;
class RockBand {
  public:
    RockBand(string a_name, string a_leadSinger, int a_yearFounded){
      //Implement the Constructor here
    }

  friend ostream& operator<<(ostream& os, const RockBand& obj){
    //Implement the output operator here
  }

  private:
    string name;
    string leadSinger;
    int yearFounded;
  };

int main() {
   //Create two RockBand objects with the values provided and display them
}
```

Figure 4: An exercise generated using our LOD-based AIG approach

References

[1] Robert K Atkinson et al. “Learning from examples: Instructional principles from the worked examples research”. In: *Review of educational research* 70.2 (2000), pp. 181–214.

[2] Christian Bizer, Tom Heath, and Tim Berners-Lee. “Linked data-the story so far”. In: *Linking the World’s Information: Essays on Tim Berners-Lee’s Invention of the World Wide Web*. 2023, pp. 115–143.

[3] William W Cobern. "Contextual constructivism: The impact of culture on the learning and teaching of science". In: *The practice of constructivism in science education.* Routledge, 2012, pp. 51–69.

[4] Marija Cubric and Milorad Tosic. "Towards automatic generation of e-assessment using semantic web technologies". In: *International Journal of e-Assessment* (2011).

[5] Anna Eckerdal et al. "Putting threshold concepts into context in computer science education". In: *ACM Sigcse Bulletin* 38.3 (2006), pp. 103–107.

[6] Muriel Foulonneau. "Generating educational assessment items from linked open data: The case of DBpedia". In: *The Semantic Web: ESWC 2011 Workshops: ESWC 2011 Workshops, Heraklion, Greece, May 29-30, 2011, Revised Selected Papers 8.* Springer. 2012, pp. 16–27.

[7] Mark J Gierl and Hollis Lai. "The role of item models in automatic item generation". In: *International journal of testing* 12.3 (2012), pp. 273–298.

[8] Mark Guzdial and Judy Robertson. "Too much programming too soon?" In: *Communications of the ACM* 53.3 (2010), pp. 10–11.

[9] Simon Holland, Robert Griffiths, and Mark Woodman. "Avoiding object misconceptions". In: *Proceedings of the twenty-eighth SIGCSE technical symposium on Computer science education.* 1997, pp. 131–134.

[10] I-Han Hsiao, Peter Brusilovsky, and Sergey Sosnovsky. "Web-based parameterized questions for object-oriented programming". In: *E-Learn: World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education.* Association for the Advancement of Computing in Education (AACE). 2008, pp. 3728–3735.

[11] Marcia C Linn and Michael J Clancy. "The case for case studies of programming problems". In: *Communications of the ACM* 35.3 (1992), pp. 121–132.

[12] Andreas Papasalouros, Konstantinos Kanaris, and Konstantinos Kotis. "Automatic Generation Of Multiple Choice Questions From Domain Ontologies." In: *e-Learning* 1 (2008), pp. 427–434.

[13] Ferran Prados et al. "Automatic generation and correction of technical exercises". In: *International conference on engineering and computer education: Icece.* Vol. 5. 2005.

[14] Noa Ragonis and Mordechai Ben-Ari. "A long-term investigation of the comprehension of OOP concepts by novices". In: (2005).

[15] John R Savery and Thomas M Duffy. “Problem based learning: An instructional model and its constructivist framework”. In: *Educational technology* 35.5 (1995), pp. 31–38.

[16] John Sweller. “Cognitive load during problem solving: Effects on learning”. In: *Cognitive science* 12.2 (1988), pp. 257–285.

[17] John Sweller, Jeroen JG Van Merrienboer, and Fred GWC Paas. “Cognitive architecture and instructional design”. In: *Educational psychology review* 10 (1998), pp. 251–296.

[18] Ellampallil Venugopal Vinu and P Sreenivasa Kumar. “Improving large-scale assessment tests by ontology based approach”. In: *The Twenty-Eighth International Flairs Conference.* 2015.

[19] Maha Al-Yahya et al. “Ontology-based multiple choice question generation”. In: *The Scientific World Journal* 2014 (2014).

[20] Laura Zavala. “Read, Manipulate, and Write: A study of the role of these cumulative skills in learning computer programming”. In: *Proceedings of the ASEE NE 2016 Conference.* 2016, pp. 28–30.

[21] Laura Zavala and Benito Mendoza. “On the use of semantic-based aig to automatically generate programming exercises”. In: *Proceedings of the 49th ACM technical symposium on computer science education.* 2018, pp. 14–19.

[22] Laura Zavala and Benito Mendoza. “Precursor skills to writing code”. In: *Journal of Computing Science in Colleges* 32.3 (2017), pp. 149–156.

Why Neurodivergent Tech Students are Overlooked for Jobs and How Educators and Employers Can Help*

Susan S Conrad and Diane R Murphy
School of Technology and Innovation
Marymount University
Arlington, VA 22007
{sconrad,dmurphy}@marymount.edu

Abstract

Creating learning environments which are diverse, equitable and inclusive (DEI) enhance student learning and help all students transition effortlessly to the workforce. One category of students often overlooked for professional employment are neurodivergent students. These individuals are often perceived as intellectually impaired; however, they are not. Rather they simply process information and/or socially interact differently, sometimes demonstrating great strength in areas of mathematics, ideation and memory. Being neurodivergent may make college more difficult and hinder their ability to find meaningful employment in their chosen field. Awareness of neurodivergent nuances provides a framework for educators and employers to adopt strategies that build upon the unique strengths of neurodivergent individuals to maximize their success in college and in the workplace. This paper describes a small pilot study of the challenges neurodivergent individuals face and proposes some best practices that educators and employers can use to enable neurodivergent students to succeed in college and the workforce.

1 Introduction

What do Pokeman, Symphony 40 in G minor and Virgin Airlines all have in common? The answer is that all the creators, Satoshi Tajiri, Wolfgang Amadeus Mozart, and Richard Branson, are neurodivergent thinkers who look at the world through a unique lens. The terms *neurodivergent* and *neurodiversity* were coined in the 1990s by Judy Singer, a sociologist with autism, who wanted to give a voice to people with different brain wiring by recognizing cognitive variations between individuals [4]. She stressed the importance that neurodiver gent individuals be given the same privileges as a social category. on par with gender, ethnicity, sexual orientation, or disability status. These differences might affect the way a person thinks, reads, moves, interacts with others, or processes information, but they do not make these individuals less effective or intelligent [7].

Statistics greatly vary as to how many people may be neurodivergent, with numbers ranging from 16% to as high as 40% of the population [14]. The difficulty with identifying the exact number is complex , with issues such as what conditions and/or diagnoses are considered to fall in the category of neurodivergent; individual preferences to disclosing such personal information; lack of understanding by many people; societal prejudice for anything seen as not “normal"; and cultural bias [5]. However , there is a shift on the horizon as neurodivergent individuals are beginning to be recognized for the unique abilities they bring, such as attention to detail, strong memory retention, persistence and performance with repetitive tasks [15].These traits make these individuals ideal for positions within many workforce pathways, such as technology, engineering, manufacturing, finance and accounting [10].

Technology giants such as Microsoft, IBM, HP, Google, VMWare and Dell are just a few of the companies that report seeking out and hiring neurodivergent technology talent [3].In 2021, for example, Google announced that it planned to partner with the Stanford Neurodiversity project to train 500 Google managers on how to hire, orient and manage neurodivergent individuals as part of an initiative to expand the neurodivergent employee base at Google [8].

Creating safe learning spaces for neurodivergent students to thrive align with the educational goals of most institution, with the increased focus on preparing students for employment and economic mobility. However, the challenge in meeting these goals lies with understanding what neurodivergent individuals need to be successful and how to implement those accommodations into the stream of everyday student interaction, without them being “labelled" as they often have been in the K-12 educational system.

2 Background

People who think and process information in an expected way for their culture and setting are viewed as neurotypical. Some common characteristics associated with neurotypical individuals include: [4] reaching developmental milestones at a similar time to other children, such as learning to speak; (2) having social or organizational skills that are similar to peers; (3) being able to tolerate some sensory discomfort, such as loud noises without much difficulty; (4) being able to adapt to changes in routines; [15] being able to focus in class or at work for prolonged periods; (6) having varied interests or hobbies typical for the person's age [14].

In contrast, neurodivergent individuals, may be able to master some of these milestones, but struggle with others. Although different from neurotypical behaviors, neurodiversity demonstrates an evolutionary change in the human gene pool and is a completely normal biological process [2]. Thinking different ly does not mean thinking less and can be a strength in today's ever-changing world. Many neurodivergent people have strengths such as: (1) attention to detail; (2) excellent pattern recognition; (3) logical thinking; and (4) strong focus even on repetitive tasks [16].

These talents are especially beneficial for technology and engineering professions as this work requires problem-solving, attention to detail, and creativity. Many of these positions also involve structured work environments with well-defined policies and processes [10].

Neurodiversity is inclusive of a wide range of conditions ranging in severity from total dependence to highly intelligent with minimal maladies. A host of acronyms attempt to capture the vastness of conditions such as Autism Spectrum Disorder (ASD), Sensory Processing Disorder (SPD), Highly Sensitive Disorder (HSD), and others. In addition, individuals with the following conditions may also be included in the neurodiverse population.

- **ADHD/ADD:** People with Attention Deficit Hyperactivity Disorder (ADHD) have high levels of energy and have difficulty sitting still at school or work. They can also be impulsive, lack filters, and may interrupt others People with Attention Deficit Disorder (ADD) are similar to those with ADHD but do not exhibit the "hyperactivity" component associated with ADHD. Both groups can easily lose focus, have difficulty paying attention for extended periods of time and may lack conventional organizational skills. [6].
- **Learning disabilities:** Learning disabilities (LD) affect how someone learns or takes in information. There are 13 categories protected under the American with Disabilities Act. A sampling include: (a) Dyslexia affects a person's ability to read; (b) Dysgraphia affects handwriting and

fine motor skills; (c) Dyscalculia impacts learning of numbers and facts. (d) Non-verbal LD affects a person's ability to interpret nonverbal cues like facial expressions or body language and may have poor coordination. Individuals with these diagnoses frequently have average or above-average intelligence but require alternative methods to learn based upon their LD. LD can also interfere with higher level skills such as organization, time planning, abstract reasoning, long or short-term memory and attention. [13]

- **Dyslexia:** Difficulty with reading, writing, spelling, and listening. These individuals often show a preference for non-linear thinking. [12]
- **Autism:** Autism affects how someone processes sensory information and how they think and communicate. For example, an autistic individual may have tactile sensitivity, noise sensitivity, fear of crowds, or have trouble reading social cues. Others may have very specific, in-depth interests. The effects vary greatly from person to person. [11]
- **Tourette's syndrome:** People with Tourette's syndrome have a neurological condition that causes them to make involuntary movements or sounds, known as tics which a person has no control over. It can be exasperated by physical, emotional or mental stress, [17].
- **Synesthesia:** https://en.wikipedia.org/wiki/SynesthesiaSynesthesia is a neurological phenomenon in which stimulation of one sense leads to automatic, involuntary experiences of a second one. For example a person might see colors while listening to music or feel the touching experience they see or hear about from another person [9].
- **Sensory Processing Disorders (SPD):** People with SPD are unable to modulate and integrate sensory stimuli. They can become overstimulated and anxious, not reacting to the stimuli in a neurotypical manner [1].

As evident from the above non-exhaustive list of naturally occurring human neurological variations, helping students to learn requires sensitivity to the needs of all students. Neurodivergent individuals behave and experience the world in different ways because their brain processes information and sensory inputs in a way different from neurotypical people[1]. Since there is a great span of diagnosed categories, it is difficult to address every specific need. However, there are some common threads between all conditions which educators and employers can implement to assist neurodivergent students learn and work: the goal of the research study.

3 Methodology

As long-time educators in a university information technology program, we have personally noted the behaviors and difficulties of several neurodivergent male and female students. Of particular note has been their transition from a successful college career to being hired as a professional. As we see the neurodivergent problem growing, we sought to find what more can be done to level their playing field. We know they can provide excellent problem solving by thinking outside of the box, so needed in fields such as cybersecurity. While we have supported a few students into their dream job, we know more has to be done.

We began this journey by questioning a small set of neurodivergent students. An explorative study, using a qualitative semi-structured interview method. was used to obtain data from eight participants. The protocol was submitted to the university's Institutional Review Board (IRB) and approved as exempt (ID 826 as all participants were over the age of 18).

The interview candidates self-identified with one or more of the neurological patterns discussed in the previous section and came from the U.S. The interviews were conducted in one of the following the modes: in-person, zoom, phone and written. The individuals were asked 13 questions about their experiences with attending college and transitioning to the workplace. The interviews were transcribed by one of the authors using inductive qualitative methods resulting in the data being coded and organized into themes.

4 Findings

Overall, the participants said that making it through school and finding a job required them to work very hard to meet the expectations of educators and employers. All but one participant had graduated from college. Seven were employed in professional jobs. Three were in technology related positions, and the others were in healthcare, food industry, human resources, and administration. The participants were asked what made them neurodiverse. Table 1 provides a breakdown of the participant demographics. It should be noted that several of the participants indicated to have multiple neurodivergent diagnoses but the table reflects the individual's self-identified primary diagnosis.

Table 1: **Table Participant Analysis**

Neurodivergent Category	**Sex**	**Number of Participants**
Attention Deficit Hyperactivity	M	1
Attention Deficit	F	2
Autistic	M	2
Autistic	F	2
Learning Disability	M	1

Although anxiety disorder is not officially a neurodivergent condition, five of the participants said they have persistent anxiety and stress. One of the autistic participants is also dyslexic. When each participant was asked when they knew she was was neurodivergent. Six said they were diagnosed as children but two did not know until they were adults. When asked about their time in school seven participants noted that were had been bullied by peers and had many negative experiences in their middle and high school years. Six said teachers labeled them and put in "special" groups away from the rest of the class which strongly impacted their social relationships with other students. Of those attending post-secondary education, only two told anyone at their educational institution about being neurodiverse because they did not want to be labeled like they had been in their prior education settings.

All participants said that they relied on connections they made at their university to help them prepare for the job search. Seven of the participants said they depended heavily on their families to provide support throughout the job search process. Five of the participants said they became very discouraged and often wanted to stop looking because of lack of responses to their applications. One participant noted that they had read 85% of all autistic college graduates are either unemployed or underemployed and they did not want to be one of them. This participant told a story of not getting a position because she did not "give good eye contact". After being rejected, she contacted the interviewer for feedback and was told that it appeared she was not interested in the position and the people because her eyes wandered throughout the interview. This participant went on to say that it is "hard to give eye contact" because it is straining on the eyes and stressful for neurodivergent people. Four of the participants indicated that eye contact is very difficult.

Most of the initial job interviews are currently being carried out using virtually so anxiety about making the videoconferencing technology work was the first worry. Several participants said that they struggled with the initial small talk during the interview and usually could tell when something went wrong. Four people said that the use of sarcasm by the interviewers was lost on them and may have cost them the job. One participant said she learned “how to play the game or she wouldn’t be employed". Another person said that he learned to not talk too much and slow his speech. Another person said she felt she was discriminated because she did not answer the interview questions fast enough.

All said they did not disclose that they were neurodiverse to any potential employer during the interview and only two people disclosed it after being hired. All said they “masked" their neurodivergent conditions and tried to model neurotypical behaviors during the interview and “as best as they could" while at work. Four of the participants said that working from home has made work much easier and less stressful because they could be “myself". Several persons said that most people have stereotypes of what autism and neurodivergent are like and they have strong biases against hiring them. One participant said that “I feel discrimination every time I interview" because employers have an unconscious bias against neurodiverse persons.

Seven of the participants indicated that educating teachers and interviewers about the benefits that come from neurodivergent persons would be very helpful. One person said he attributed being denied a promotion because told his manager he was autistic. Once employed several participants said that clear directions with measurable outcomes are important for achieving the expected results. Several participants said that when given vague instructions, they became very frustrated and lost momentum with their position.

When asked what educators can do to enhance the learning experience, the participants noted that “empathy" was a good start. Two participants identified flexibility when it comes to deadlines. “For example, when I am asking for an extension, it’s not that I haven’t tried but sometimes my anxiety gets in the way." Clear instructions with examples were mentioned by all participants. “Do not assume I understand general ideas. I need details." Another stated that lecturing was a sure way for “my ADD" to take over and for them lose interest in the class. All participants mentioned a dislike for long lectures. Instead have short lectures with integrated hands-on exercises to help retain focus.

Several participants commented about their fear of presenting in front of the class. Since neurodivergent people struggle with eye contact and often cannot read the social cues of others, providing alternative methods to present individuals projects to the class are helpful. Three students discussed how one of their professors took an interest in them as individuals and how meaningful it was for them. “I really liked going to Professor X’s class because she knew me and we would talk about topics I cared about."

Those participants who identified themselves as autistic said that they have a strong sense of justice and when they see wrongdoing, they want action to be taken to rectify the situation. In school it may be reacting to students cheating or in the workforce it might be employees breaking policies. Autistic individuals rely on rules to determine the parameters of right and wrong so when they see a violation they react and want it to be remedied.

This is a summary of the findings from the interviews in our pilot study. Based upon the findings, we are making several recommendations for both educators and employers to hopefully enhance the success of neurodivergent individuals.

5 Recommendations

The authors believe that the findings from this pilot study, and their ongoing research, point to several significant implications for helping neurodivergent enhance their post-secondary learning experience as well as their employment prospects upon graduation.

One common theme that emerged from the research is the importance for educators and employers to learn what it means to be neurodivergent and understand the special talents and needs for each neurodivergent category. This will help educators and employers maximize the talents of neurodivergent employees/students, intertwining the skills or neurotypical and neurodivergent people. The hope is that as awareness about neurodivergence increases unconscious biases and discrimination will be reduced. To truly have an inclusive learning or work environment means providing accommodations for neurodivergent as well as other discriminated groups. Diversity, Equity, and Inclusion (DEI) programs must include neurodivergent individuals and provide necessary accommodations for their success while providing a safe space for expression. The American Disabilities Act protects neurodivergent individuals and as this population of people gains greater presence, educators and employers must listen or they will be in violation of the law.

Some overall recommendations that impact both educators and employees include the following suggestions: (1) Eliminate jargon in conversation; (2) Don't use sarcasm; (3) Be very clear and concise with directions; (4) Do not assume the neurodivergent persons will understand euphemisms; (5) Jokes may not be understood; (6) Be aware of unconscious bias; (7) Be direct when providing clear and consistent feedback; (8) Lack of eye contact does not mean disinterest; (9) Be aware of sensory overload – sound, lights, smells; and (10) Assign a buddy. These recommendations can help simplify the sensory complexities that neurodivergent individuals encounter in their everyday activities.

Educators can change their teaching by incorporating a few additional strategies which may benefit the entire class, not just the neurodivergent students. These include: (1) Break lectures into segments and add a hands-on activity as this is especially beneficial for students with ADD/ADHD; (2) Have students work in small groups so that social engagement is not as threatening; (3) Create alternate modes for classroom presentations; (4) Engage with all students and be respectful if the student prefers to be left alone; (5) Call students by their name; (7) Be organized and don't make a lot of changes to assignments during the course; (8) Avoid physical contact; (9) Add a clause in your syllabus telling students that you will work to help them be successful. One such statement might say: "My classroom is a safe inclusive environment for all. I am here for your success and if there are any circumstances that may impact your success in this class, let's work together to eliminate those impediments."

Employers can help neurodivergent people in both the hiring and retention process by implementing the following suggestions in the hiring process. These include: (1) Provide all candidates a list of potential questions prior to the interview, if possible; (2) Be mindful of the setting for the interview, free of distracting sounds, smells, people traffic and lighting. (3) Avoid physical contact; (4) Tell the candidates what to expect during the interview by providing an agenda, of people, topics, and timelines; (5) Ask specific questions and expect direct answers.

A good practice for all new hires is to the new employee what accommodations best help with productivity and success on the job. This will create an opportunity for learning more about the new team member and demonstrate concern for the individual. Some suggestions that can be implemented which will benefit both neurotypical and atypical individuals include: (1) Assign a buddy; (2) Make sure working accommodations are acceptable; (3) Update the team about the new employee and provide sensitivity training if acceptable to the new employee; (4) Maintain regular one-on-one meetings with the new employee; (5) Ask for feedback and suggestions for improvement; (6) Act quickly on those improvements suggested; (7) Set boundaries regarding deadlines, work hours, work products and team collaborations and (8) Stay engaged with all employees.

6 Conclusion and Future Research

Creating a diverse, equitable and inclusive classroom or workplace are values that strive to support people of different genders, sexual orientations, ethnicities, religions, and disabilities. However, to be fully inclusive, neurodiversity must also be a part of the organization's DEI strategy. Most people who are neurodivergent are not disabled, but rather think and behave differently. Unfortunately, many very talented neurodivergent are unemployed or underemployed because of unfair hiring practices, discrimination, and bias.

This study was a pilot study to better understand the issues facing neurodivergent students as they transition to the workforce from these individuals themselves. It provides an initial framework for creating successful environments that will help neurodivergent people thrive and fully utilize their talents.

This research is the first of three studies to be conducted by the authors to understand how to enhance the job success for neurodivergent individuals. Since there is growing interest in making a difference for neurodivergent individuals, future research will concentrate on connecting specific neurodivergent benefits with specific career options. Research will also focus on training managers and recruiters how to recruit and retain neurodivergent employees utilizing fair, non-discriminating procedures. Since several companies as described in this paper's introduction section have already created neurodiverse programs and policies, research to evaluate the success of such programs is another area of study. Lastly researching ways to empower neurodivergent individuals to advocate for themselves as either a group or individually is one of the most impactful ways to bring about meaningful change and create a neurodiverse environment benefiting all.

References

[1] K. Mcmahon et al. *A Path From Childhood Sensory Processing Disorder to Anxiety Disorders: The Mediating Role of Emotion Dysregulation and Adult Sensory Processing Disorder Symptoms.* 2019. URL: https://www.frontiersin.org/articles/10.3389/fnint.2019.00022.

[2] H. Angulo-Jiménez and L. DeThorne. "Narratives About Autism: An Analysis of YouTube Videos by Individuals Who Self-Identify as Autistic". In: *Am. J. Speech - Lang. Pathol. Online* 28.2 (May 2019), pp. 569–590.

[3] Austin and Pisano. *Neurodiversity as a Competitive Advantage.* May 2017. URL: https://hbr.org/2017/05/neurodiversity-as-a-competitive-advantage.

[4] N. Baumer and J. Freuh. *What is neurodiversity?* 2021. URL: https://www.health.harvard.edu/blog/what-is-neurodiversity-202111232645 (visited on 11/23/2021).

[5] E. Bergman. *Autistic Individuals Are Still Overlooked For Work.* Apr. 2022. URL: https://www.autismparentingmagazine.com/autistic-overlooked-work-opportunities.

[6] CDC. *Symptoms and Diagnosis of ADHD.* 2020. URL: https://www.cdc.gov/ncbddd/adhd/diagnosis.html (visited on 06/24/2022).

[7] A. Cooks-Campbell. *Why you shouldn't overlook neurodiversity in your DEI strategy.* 2022. URL: https://www.betterup.com/blog/neurodiversity (visited on 01/07/2022).

[8] R. Enslin. *Google Cloud launches a career program for people with autism.* 2021. URL: https://cloud.google.com/blog/topics/inside-google-cloud/google-cloud-launches-a-career-program-for-people-with-autism.

[9] Martino G and Marks L. "Synesthesia: Strong and Weak". In: *Curr. Dir. Psychol. Sci* 10 (2 Apr. 2001), pp. 61–65. DOI: 10.1111/1467-8721.00116.

[10] A. Green. *5 Neurodiversity-Friendly Career Paths.* Jan. 2020. URL: https://differentbrains.org/5-neurodiversity-friendly-career-paths.

[11] Nerenberg J. *Divergent Mind.* HarperOne, 2020.

[12] C. von Károlyi et al. "Dyslexia linked to talent: Global visual-spatial ability". In: *Brain Lang.* 85.3 (June 2003), pp. 427–431.

[13] LDA. *Types of Learning Disabilities.* June 2022. URL: https://ldaamerica.org/types-of-learning-disabilities (visited on 06/24/2022).

[14] Lerner M. *What does neurotypical and neurodivergent mean?* 2022. URL: https://www.medicalnewstoday.com/articles/what-does-neurotypical-mean (visited on 02/04/2022).

[15] Mizar S. *Neurodiversity and tech: a win-win equation.* Apr. 2021. URL: https://www.idgconnect.com/article/3614940/neurodiversity-and-tech-a-win-win-equation.html.

[16] Terkel. *12 neurodiversity strengths that come from thinking differently.* 2021. URL: https://www.texthelp.com/resources/blog/12-neurodiversity-strengths-that-come-from-thinking-differently.

[17] Braezer Y. *Tourette syndrome (TS): Symptoms, causes, and treatment.* Nov. 2021. URL: https://www.medicalnewstoday.com/articles/175009 (visited on 06/25/2022).

From Predicting MMSE Scores to Classifying Alzheimer's Disease Detection & Severity*

Saurav K. Aryal, Ujjawal Shah, Legand Burge, Gloria Washington
Department of Electrical Engineering and Computer Science
Howard University
Washington, DC 20059, USA
{saurav.aryal, lburge, gloria.washington}@howard.edu
{ujjawal.shah}@bison.howard.edu

Abstract

In this paper, we perform predictive modeling of Mini-Mental State Examination (MMSE) scores using four operationally different standard machine learning (ML) models (kNN, SVM, linear, & lightgbm) while utilizing a combination of acoustic-linguistic features and hyper-parameter optimization algorithms. Our proposed approach outperforms the current best-performing research model with a 25% drop in Root Mean Square Error (RMSE). The regression model outputs are then thresholded to perform two classification tasks: Alzheimer's Disease (AD) Detection (AD vs. non-AD) and Severity Prediction (no dementia, early stage, and medium stage). Our best regression model outperforms task-specific models obtained via an extensive neural architecture search across all three tasks. The binary classification approach provides high accuracy (92% for binary classification and 77% for multi-class classification). We finalize our analysis with a comparative analysis of acoustic versus linguistic features and find that linguistic features are better predictors of MMSE scores.

1 Introduction

The global increase in life expectancy and the projected rise in the number of individuals with Alzheimer's Disease (AD) emphasize the need for research on aging-related diseases and the development of cost-effective technologies for detecting and tracking AD. Current diagnostic methods for AD involve surveys, psychological assessments, cognitive tests, and medical imaging, but they are limited in availability, require medical expertise, or are costly. As a result, researchers have explored non-invasive techniques such as speech and linguistic analysis as a promising approach for AD detection. This paper focuses on predictive modeling of the Mini-Mental State Examination (MMSE) using machine learning models and acoustic-linguistic features. Additionally, a novel approach is proposed for classifying AD status and stage, comparing the results to deep learning techniques. The subsequent section of the paper provides a review of relevant works in AD detection and evaluation through acoustic and linguistic analysis.

2 Relevant Works

The link between memory loss and AD-related neurodegeneration is well recognized, numerous current research focuses on linguistic and auditory abnormalities in patients. Auditory abnormalities such as dysarthria/slurring, stuttering, monotonous speech, and increased latency [7] have been identified as characteristic symptoms of AD. Additionally, linguistic features such as limited word usage and flawed grammar have also been observed in individuals with AD [11, 15]. Researchers have incorporated these features into the development of detection methods for AD. In a recent work, authors extracted the most extensive set of over 13,000 features from the same dataset [3]. [3] trained multiple linear models and found that the linear model optimized with stochastic gradient descent (SGD) on a set of the top 53 most correlated features. However, they augment their sample size by utilizing each audio chunk as a separate observation. They neither perform hyperparameter optimization nor report results averaged over multiple seeds. Despite these criticisms, they have the current best-performing for the given task-dataset pair with a training and test RMSE of 2.37 and 3.90, respectively, and made their source code available. For this regression task, we will refer to the work of [3] as the baseline and compare our models and feature sets against it henceforth.

3 Methodology

3.1 Dataset Review

We utilize the ADReSS dataset in this study to enable comparisons to the baseline. ADReSS is one of the first publicly available AD-focused data balanced by age and gender among participants, and the two specified tasks established are an MMSE score regression task and an AD classification task(AD vs. not-AD). It comprises normalized audio and manually annotated transcript data from 156 elderly participants performing the Cookie Theft task from the Boston Diagnostic Aphasia exam[13]. Using the test, a medical professional evaluated the patient with a possible score of 0-30 on the Mini-Mental Status Exam (MMSE), with lower scores of MMSE indicating a more severe cognitive decline. The test has been used across medical research literature to diagnose AD and evaluate its severity [16]. Thus, we used audio and text data for feature extraction and modeling. While utilizing the same dataset, the critical difference between our work and [3] is our smaller sample size and the choice of models.

3.2 Preprocessing, Feature Extraction, and Selection

3.2.1 Preprocessing

The normalized chunked audio files from the dataset were merged into one audio file per subject. Similarly, the text transcript was concatenated into a single string with new line characters as separators. The acoustic and linguistic features were then extracted from these files as detailed in Section3.2.2. Standardization was performed for models that require but not for feature scale-invariant models. The specific details regarding this will be detailed in Section2.3 below

3.2.2 Feature Extraction and Selection

We relied on the exhaustive previous work of the [3] for the comprehensive set of features extracted, which totaled over 13,000 features. They include:

Acoustic Features (11,659 Features): The acoustic features comprised learned and handcrafted acoustic features from normalized audio data. The learned features include articulation[17, 14], phonation[17, 2], and prosody[6, 17] extracted from pre-trained models. The handcrafted features include spectral, Mel frequency cepstral coefficients, and Chroma vector/deviation features totaling 138 features [8] each across 80+ combinations of frame overlaps and duration.

Linguistic Features (1,693 Features): Linguistic features include Word/ Sentence Count, Vocab Set, readability score, emotion analysis, and more[3].

These features were based on transcript files and totaled 1,693 features.

Feature selection is essential to mitigate the curse the dimensionality. We followed and relied on the extensive feature size analysis from [3] and retrieved the top 53 most-correlating features. Using the same feature set also enables direct and explicit comparison of our findings. As outlined below, the modeling and evaluation protocol utilizes this set of features. We refer interested readers to the Appendix of [3] as a resource for the extensive detail regarding the 13,000+ features involved and libraries used.

3.3 Modeling and Evaluation Protocol

3.3.1 Model Choices

We employed a diverse set of five machine learning models for predictive modeling, including Light Gradient Boosted Machine (lightGBM) [10], k-nearest neighbors (kNN), Kernel-based Support Vector Machine (SVM), linear model with SGD optimization. These models were selected based on their established applications in healthcare [5]. Additionally, we included the AutoML approach, using open-source autokeras [9] library to explore task-specific deep learning architectures. Overall, we utilized five different modeling approaches across our feature set to maximize predictive performance.

3.3.2 Training and Hyperparmeter Optimization

All models are trained on the provided training set for the MMSE prediction regression task. The kNN, SVM, linear, and lightGBM models utilize the described feature set in Section3.2.2. To determine the optimal model with optimized hyperparameters, autokeras is run over 150 trials on the feature set. Additionally, three Autokeras models are trained; 2 for AD detection and 1 AD stage classifier. A preliminary experiment is conducted on all 13,000+ features to assess the predictive significance of audio and linguistic features. Audio and text features are separated, and dimensionality reduction is performed using PCA to retain principal components that capture 90% of the feature variation. The best-performing models from the previous task are separately trained on audio and text for regression, followed by classification tasks using their outputs. Other models undergo hyperparameter search using Bayesian and Randomized algorithms across 100 iterations with Leave One Subject Out (LOSO) Cross-Validation since LOSO is prevalent in this task [12, 4, 3]. It is important to note that the hyperparameters of the models in [3] were not tuned. Hence, a total of different models are obtained.

3.3.3 Evaluation

The models are trained for the specified regression task and evaluated using RMSE, the standard metric for this task-dataset pair [12]. We compare the top-performing models from each of the five types (SVM, kNN, lightGBM, linear, autokeras) with the baseline's reported performance on both training and testing data.

Explicit labels for the AD detection and AD severity tasks were not available in the dataset. Due to the lack of consensus in medical research literature, we applied multiple thresholds to label participants for AD detection and severity classification. The thresholds used for labeling AD severity are presented in Table 4. Standard classification metrics (precision, recall, accuracy, f1) were calculated for each approach in the three classification experiments.

Table 1: Multiclass Classification Threshold

Class	**MMSE Score**
No-Dementia	≥ 25
Early Stage Dementia	≥ 19 and < 25
Moderate-Dementia	≥ 10 and ≤ 20
Severe Impairment	≤ 9

4 Results

4.1 MMSE score Regression

The RMSE scores for the hyperparameter-optimized models and the Autokeras model in the regression task are available in Table 2. Despite a substantial difference in sample sizes, we achieved similar baseline performance on both the training and test sets using the same model and features. Among the models, only lightgbm outperformed the baseline [3] on the training set, with a slight improvement of 0.30. However, on the test set, all conventional machine learning models (linear, SVM, kNN, & lightgbm) outperformed the baseline [3]. Notably, lightgbm exhibited a significant improvement, reducing the test RMSE by over 25% compared to the baseline. Despite being newer advancements, Autokeras' deep learning architecture did not achieve comparable performance. This outcome suggests that such large models may not always be suitable for low-resource modeling problems and may require specific adaptation. Additionally, Bayesian optimization demonstrated superior performance to randomized search for hyperparameter tuning.

Figure 1: Box Plot of Residuals for Baseline Models

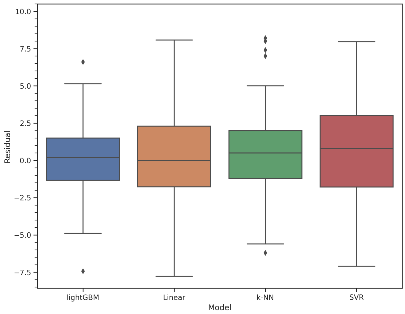


Table 2: RMSE Scores

Model	HyperParameters	Train RMSE	Test RMSE
lightgbm	Bayesian Optimization	2.28	2.87
	RandomizedSearchCV	0.02	3.96
Linear	Bayesian Optimization	3.85	3.62
	RandomizedSearchCV	3.78	5
k-NN	Bayesian Optimization	3.46	3.55
	RandomizedSearchCV	5.33	4.32
SVR	Bayesian Optimization	2.31	3.63
	RandomizedSearchCV	8.1	6.64
Autokeras		6.94	8.08

We further evaluate all Bayesian-optimized models using the boxplot of test residuals in Figure 1. LightGBM and kNN exhibit better predictions based on their residual distributions. LightGBM's predictions fall within the Inter Quartile Range (IQR) of [1.5, -1.5], while kNN's IQR is [-1.5, 2.0]. We expect that lightGBM will perform slightly marginally than kNN in most cases. Despite kNN's higher average RMSE, it also performs noticeably worse than the lightGBM model overall and on the remaining 50% of predictions.

4.2 Binary Classification - AD Detection

Binary Classification results from the three labeling approaches for AD detection (MMSE thresholds of 24, 25, and 26) yield varying outcomes & are summarized in tables (Table 3, Table 4, and Table 5). Overall, lightgbm performs the best across all thresholds, approaching the performance of the current best model [18] at thresholds 24 and 26, but with smaller training time and model size. At threshold 25, lightGBM, kNN, and Autokeras achieve equal accuracy. However, the task-specific classifier model performs notably worse compared to the best model outside this threshold.

Table 3: Classification Report for best-performing regression models for binary classification with MMSE Threshold 24

Model	Categories	precision	recall	f1-score	support	accuracy
lightGBM	**Dementia**	**0.81**	**0.94**	**0.87**	**18**	**0.90**
	No-Dementia	**0.96**	**0.87**	**0.91**	**30**	
Linear	Dementia	0.72	1	0.84	18	0.85
	No-Dementia	1	0.77	0.87	30	
k-NN	Dementia	0.75	1	0.86	18	0.88
	No-Dementia	1	0.8	0.89	30	
SVR	Dementia	0.72	1	0.84	18	0.85
	No-Dementia	1	0.77	0.87	30	
Autokeras	Dementia	0.72	1.0	0.83	18	0.85
	No-Dementia	1	0.76	0.86	30	

Table 4: Classification Report for best-performing regression models for binary classification with MMSE Threshold 25

Model	Categories	precision	recall	f1-score	support	accuracy
lightGBM	**Dementia**	**0.82**	**0.9**	**0.86**	**20**	**0.88**
	No-Dementia	**0.92**	**0.86**	**0.89**	**28**	
Linear	Dementia	0.76	0.95	0.84	20	0.85
	No-Dementia	0.96	0.79	0.86	28	
k-NN	**Dementia**	**0.79**	**0.95**	**0.86**	**20**	**0.88**
	No-Dementia	**0.96**	**0.82**	**0.88**	**28**	
SVR	Dementia	0.73	0.95	0.83	20	0.83
	No-Dementia	0.95	0.75	0.84	28	
Autokeras	**Dementia**	**0.82**	**0.90**	**0.86**	**20**	**0.88**
	No-Dementia	**0.92**	**0.86**	**0.89**	**28**	

Table 5: Classification Report for best-performing regression models for binary classification with MMSE Threshold 26

Model	Categories	precision	recall	f1-score	support	accuracy
lightGBM	**Dementia**	**0.87**	**0.95**	**0.91**	**21**	**0.92**
	No-Dementia	0.96	0.89	0.92	27	
Linear	Dementia	0.8	0.95	0.87	21	0.88
	No-Dementia	0.96	0.81	0.88	27	
k-NN	Dementia	0.83	0.95	0.89	21	0.90
	No-Dementia	0.96	0.85	0.9	27	
SVR	Dementia	0.77	0.95	0.85	21	0.85
	No-Dementia	0.95	0.78	0.86	27	
Autokeras	Dementia	0.77	0.95	0.85	21	0.85
	No-Dementia	0.95	0.78	0.86	27	

4.2.1 Ternary Classification - AD Severity

We assessed the dataset and models' ability to identify different severities of Alzheimer's disease (AD). For three-class classification, we applied the same methodology as the binary classification task since severe AD cases were absent in the dataset. The results (Table 6) indicated little or no significant difference among the models. SVM performed the best, while lightgbm and autokeras showed poorer performance. All models excelled in non-AD class classification but faced challenges in early and mild AD classification. These findings align with observations concerning MMSE and mild-early stage AD [1].

Table 6: Multiclass Classification for best-performing regression models

Model	Categories	precision	recall	f1-score	support	accuracy
lightGBM	Early Stage Alzheimer's	0.11	0.17	0.13	6	0.73
	Moderate Dementia	0.77	0.71	0.74	14	
	No-Dementia	0.92	0.86	0.89	28	
K-NN	Early Stage Alzheimer's	0.12	0.17	0.14	6	0.75
	Moderate Dementia	0.71	0.71	0.71	14	
	No-Dementia	0.96	0.89	0.93	28	
linear	Early Stage Alzheimer's	0.14	0.17	0.15	6	0.75
	Moderate Dementia	0.71	0.86	0.77	14	
	No-Dementia	0.96	0.82	0.88	28	
SVR	**Early Stage Alzheimer's**	**0.25**	**0.33**	**0.29**	**6**	**0.77**
	Moderate Dementia	**0.73**	**0.79**	**0.76**	**14**	
	No-Dementia	**0.96**	**0.86**	**0.91**	**28**	
Autokeras	Early Stage Alzheimer's	0	0	0	7	0.73
	Moderate Dementia	0.59	0.93	0.72	14	
	No-Dementia	0.92	0.81	0.86	27	

4.3 Audio vs Text Analysis

To assess the impact of audio and linguistic features separately, we conducted separate analyses for each set using Principal Component Analysis (PCA). The original set of 13,000+ features was reduced to 43 components, capturing over 90% of the variance. Additionally, we tested feature sizes of 30 and 40 components to evaluate their influence. Figure 2 shows the cumulative variance explained by PCA for audio and text features. Using these components, we trained lightGBM and kNN models, presenting the results in Table 7. Our findings indicate that more audio features may be needed for the regression task, while a smaller feature size performed better for text features, suggesting potential for dimensionality reduction to be a better predictor in the future.

Table 7: Audio & Text Features

	Model	**number of principal components**	**variance explained (%)**	**Train RMSE**	**Test RMSE**
Audio Features	lightGBM	40	89.11	6.14	6.39
		30	84.23	5.83	6.39
		43	90.21	5.09	6.44
	k-NN	**40**	**89.11**	**6.25**	**6.34**
		30	84.23	6.38	6.86
		43	90.21	6.34	6.47
Text Features	lightGBM	40	89.11	0.08	4.78
		30	84.23	1.45	4.13
		43	90.21	5.09	6.44
	k-NN	40	89.11	4.08	3.79
		30	**84.23**	**3.76**	**3.77**
		43	90.21	3.34	4.49

Audio features do poor on the regression task than linguistic features and thus may have less predictive significance. These results align with the observations of [4, 12, 3]. However, exhaustive future research is needed to quantify these interdependencies.

Finally, we utilized audio and linguistic feature-based models for binary (thresholds 25/26) and ternary classifications. Table 8 presents the resulting classification metrics for both isolated features. While audio features performed poorly, text data also exhibited a noticeable drop in performance measures. Currently, textual features show superiority for the tasks in this dataset.

Figure 2: Cumulative variance explained by PCA for audio and text features

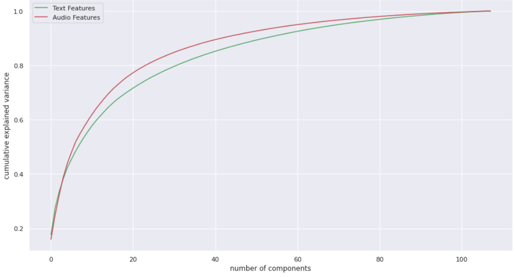


Table 8: Classification report for best performing model on audio and text features

	Classification Type	Categories	precision	recall	f1-score	support	accuracy
Audio Features	Binary Classification	Dementia	0.4	0.95	0.57	20	0.4
	MMSE Threshold -> 25	No-Dementia	0	0	0	28	
	Binary Classification	Dementia	0.44	1	0.61	21	0.44
	MMSE Threshold -> 26	No-Dementia	0	0	0	27	
	Multiclass Classification	Early Stage Alzheimer's	0.13	0.86	0.22	7	0.12
		Moderate Dementia	0	0	0	14	
		No-Dementia	0	0	0	27	
Text Features	Binary Classification	Dementia	0.72	0.9	0.8	20	0.81
	MMSE Threshold -> 25	No-Dementia	0.91	0.75	0.82	28	
	Binary Classification	Dementia	0.7	0.9	0.79	21	0.79
	MMSE Threshold -> 26	No-Dementia	0.9	0.7	0.79	27	
	Multiclass Classification	Early Stage Alzheimer's	0.25	0.43	0.32	7	0.69
		Moderate Dementia	0.71	0.71	0.71	14	
		No-Dementia	0.91	0.74	0.82	27	

5 Limitation and Future Work

Results from [18] indicate that ASR transcripts are sufficient for AD detection. However, ASR has issues with non-native, accented, and non-English speakers. While research has shifted to audio-only challenges, transcript data still performs better. The dataset did not include severe cases of Alzheimer's disease, so the detection and diagnosis of this condition may be less accurate for severe cases. Text features suggest the need for extensive research on audio versus text features. Regression models perform adequately but do not allow for evaluating Precision-Recall tradeoffs and other important measures. Reported performance is an average, so exploring predictive performance with confidence intervals across different random states or seeds would be valuable. This research is not a substitute for medical professionals but can assist caregivers. Limited and constrained data requires further work before real-world implementation.

6 Conclusion

We used four machine learning models to predict Mini-Mental State Examination (MMSE) scores from acoustic-linguistic features. Our approach outperformed the current best-performing model with a 25% drop in RMSE. We then used the regression model outputs to perform two classification tasks: Alzheimer's Disease (AD) detection and severity prediction (no dementia, early stage, and medium stage). Our best regression model outperformed task-specific models obtained via an extensive neural architecture search across all three tasks. The binary classification approach provided high accuracy (92% for binary classification and 77% for multi-class classification). We finalize our analysis with a comparative analysis of acoustic versus linguistic features and find that linguistic features are better predictors of MMSE scores.

References

[1] I. Arevalo-Rodriguez, N. Smailagic, M. R. i Figuls, A. Ciapponi, E. Sanchez-Perez, A. Giannakou, O. L. Pedraza, X. B. Cosp, and S. Cullum. Mini-mental state examination (mmse) for the detection of alzheimer's disease and other dementias in people with mild cognitive impairment (mci). *Cochrane database of systematic reviews*, (3), 2015.

[2] T. Arias-Vergara, J. C. Vásquez-Correa, and J. R. Orozco-Arroyave. Parkinson's disease and aging: analysis of their effect in phonation and articulation of speech. *Cognitive Computation*, 9(6):731–748, 2017.

[3] S. K. Aryal, H. Prioleau, and L. Burge. Acoustic-linguistic features for modeling neurological task score in alzheimer's. In *PACIFIC SYMPOSIUM ON BIOCOMPUTING 2023: Kohala Coast, Hawaii, USA, 3–7 January 2023*, pages 335–346. World Scientific, 2022.

[4] A. Balagopalan, B. Eyre, F. Rudzicz, and J. Novikova. To bert or not to bert: comparing speech and language-based approaches for alzheimer's disease detection. *arXiv preprint arXiv:2008.01551*, 2020.

[5] A. Callahan and N. H. Shah. Machine learning in healthcare. In *Key Advances in Clinical Informatics*, pages 279–291. Elsevier, 2017.

[6] N. Dehak, P. Dumouchel, and P. Kenny. Modeling prosodic features with joint factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(7):2095–2103, 2007.

[7] I. Ferrer, A. Aymami, A. Rovira, and J. M. Grau Veciana. Growth of abnormal neurites in atypical Alzheimer's disease. *Acta Neuropathologica*, 59(3):167–170, Sept. 1983.

[8] T. Giannakopoulos. pyaudioanalysis: An open-source python library for audio signal analysis. *PloS one*, 10(12), 2015.

[9] H. Jin, F. Chollet, Q. Song, and X. Hu. Autokeras: An automl library for deep learning. *Journal of Machine Learning Research*, 24(6):1–6, 2023.

[10] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.

[11] J. O. d. Lira, T. S. C. Minett, P. H. F. Bertolucci, and K. Z. Ortiz. Analysis of word number and content in discourse of patients with mild to moderate alzheimer's disease. *Dementia & neuropsychologia*, 8:260–265, 2014.

[12] S. Luz, F. Haider, S. de la Fuente, D. Fromm, and B. MacWhinney. Alzheimer's dementia recognition through spontaneous speech: The adress challenge. *arXiv preprint arXiv:2004.06833*, 2020.

[13] B. MacWhinney. The childes project: Tools for analyzing talk: Volume i: Transcription format and programs, volume ii: The database, 2000.

[14] J. R. Orozco-Arroyave, J. C. Vásquez-Correa, J. F. Vargas-Bonilla, R. Arora, N. Dehak, P. S. Nidadavolu, H. Christensen, F. Rudzicz, M. Yancheva, H. Chinaei, et al. Neurospeech: An open-source software for parkinson's speech analysis. *Digital Signal Processing*, 77:207–221, 2018.

[15] F. Rudzicz, G. Hirst, P. van Lieshout, G. Penn, F. Shein, A. Namasivayam, and T. Wolff. Torgo database of dysarthric articulation, 2012.

[16] K. Schultz-Larsen, R. K. Lomholt, and S. Kreiner. Mini-mental status examination: a short form of mmse was as accurate as the original mmse in predicting dementia. *Journal of clinical epidemiology*, 60(3):260–267, 2007.

[17] J. C. Vásquez-Correa, J. Orozco-Arroyave, T. Bocklet, and E. Nöth. Towards an automatic evaluation of the dysarthria level of patients with parkinson's disease. *Journal of communication disorders*, 76:21–36, 2018.

[18] Y. Wang et. al. Exploring linguistic feature and model combination for speech recognition based automatic ad detection. *arXiv preprint arXiv:2206.13758*, 2022.

Term Frequency Features vs Transformers: A Comparision for Sentiment Classification of African Languages*

Saurav K. Aryal, Howard Prioleau, Ujjawal Shah
Sameer Acharya
Department of Computer Science
Howard University
Washington, DC 20059
saurav.aryal@howard.edu

Abstract

Severe limitations in data and technological availability have vastly affected NLP research into African languages. With Africa having over 2000 languages, the lack of NLP research is a massive flaw within the NLP field. African languages can hold the key to the next significant advancement in NLP research because some researchers suggest that 30% of current-day languages are derived from African languages. With Sentiment Analysis being a foundational part of NLP research, the release of the AfriSenti-SemEval Shared Task 12, hosted as a part of The 17th International Workshop on Semantic Evaluation, has provided 14 new annotated datasets for Sentiment Analysis on African languages. We utilize these datasets to evaluate our approach: Delta TF-IDF features with conventional machine learning models. Delta TF-IDF results showed that our approach could provide promising results with the low resource task of sentiment analysis on African Languages. Since it utilized a significantly less data than its transformer counter parts.

1 Introduction

Although Africa is home to over 1 billion people and is one of the largest continents with vast resources, Africa's development has been hindered by climate and geopolitical issues[7], leading to a lack of infrastructure development[17]. The crucial infrastructure needed for research and development, including educational infrastructure and internet access, is inadequate in Africa, with only 22% of Africans having internet access[18]. Furthermore, Africa's current day research output is only 3% [10] of all publications, while the European Union and the United States of America produced 38.8% and 33.6% of the world publications. This is directly seen with the under-representation of African languages in technology. This limits research opportunities, particularly in Natural Language Processing (NLP). This paper focuses on NLP, specifically sentiment classification in 12 diverse African languages, to address the gap [13]. This gap hinders the applications of NLP for African languages and all languages since further analysis bodes well for overall NLP advancements.

Sentiment analysis is a crucial aspect of NLP, categorizing emotions into positive, negative, and neutral. It plays a vital role in measuring the socio-emotional impact of topics in the online world. Sentiment analysis finds applications in various fields, including education, healthcare, business, and policy-making. Additionally, it can contribute to scaling African economies through eCommerce. This paper seeks to help advance and contribute to the NLP and sentiment analysis literature for African languages by extracting Delta Term Frequency-Inverse Document Frequency (TF-IDF) features [12] and evaluating multiple machine learning models for sentiment classification.

The following section covers the utilized dataset for sentiment classification across multiple languages. Following that, we cover recent work in learning representations for multilingual modeling of African languages using deep learning. Furthermore, we explore the applicability of Delta Term Frequency-Inverse Document Frequency (TF-IDF) features for sentiment classification tasks.

1.1 Dataset - AfriSenti-SemEval / NaijaSenti

AfriSenti-SemEval [15, 20] is a recently released and landmark dataset available for African languages for sentiment analysis. Since it is hosted as a part of The 17th International Workshop on Semantic Evaluation, it provides a public platform to shed light on the sentiment analysis of African languages. At the time of writing, monolingual sentiment annotated datasets of 12 languages are made available. The task is co-created by the creators of NaijaSenti [15] and expands on their previous dataset. They provided 13 datasets comprising 12 different languages, each being a dataset and a dataset composed of all the languages. The 12 African Langauges covered are Hausa(HA), Yoruba(YO), Igbo(IG), Nigerian Pigdin(PCM), Amharic(AM), Algerian Arabic(DZ), Moroccan Arabic/Darija(MA), Swahili(SW), Kinyarwanda(KR), Twi(TWI), Mozambican Portuguese(PT), and Xitsonga(Mozambique Dialect) (TS). These dataset languages cover a wide range of different regions of Africa. Each tweet in each dataset is manually annotated between 3 positive, negative, and neutral labels. Some datasets contain code-mixed data (data that contains two or more languages) and transliteration (converting text from one script to another that involves swapping letters). While most of these languages have a limited amount of corpus, to our knowledge, some languages, such as Xitsonga, have labeled sentiment analysis datasets created for the first time [4]. Since the test labels of the datasets, as part of a competition, are yet to be released at the time of writing, current work with the entire data is limited.

1.2 Models

1.2.1 Large Multilingual Models

Large Language Models (LLMs) like XLM-RoBERTa[8] have achieved success, but they have limited support for African languages. However, there are three notable models developed specifically for African languages: AfriBERTa [16], AfroXLMR [1], and AfroLM [9]. AfriBERTa [16] demonstrated high performance in Named Entity Recognition (NER) and Text Classification tasks across 11 African languages. AfroXLMR [1] took a different approach by adapting XLM-R and achieved competitive performance in NER, news topic classification, and sentiment classification. AfroLM [9]introduced a novel approach using active learning and showed improved performance compared to other models in NER, topic classification, and sentiment classification.

Despite the success of these three models, recent work benchmarked the performance of these datasets across all languages available in the AfriSenti-Semeval dataset by utilizing only the training and validation sets. Aryal, Prioleau, and Aryal compare the mean of standard classification metrics (F-1, precision, recall, and accuracy) and find that no one-model-fits-all solution exists across the entire dataset and languages. They further train these models on all languages and find that AfroXLMR outperforms the other models. Furthermore, they provide more supporting evidence that standard XLMR does not fare well in African languages, and dedicated language or region-specific models may be preferred. We compare our results to the Aryal, Prioleau, and Aryal's benchmarks.

1.2.2 Delta TF-IDF

Delta term frequency-inverse document frequency (TF-IDF) [12] was developed to improve TF-IDF's ability on sentiment classification. TF-IDF is a statistical approach that evaluates the importance of a word to a document within a collection of documents by multiplying the times a word appears in a document and the inverse document frequency of the word across a set of documents. This formulation prevents recurring words with limited information, such as "the" and "this" from being considered important and improves the evaluation of the important terms. Delta TF-IDF improves on the traditional TF-IDF approach by weighting values by how biased they are to one document instead of[12] instead of weighting values by their rarity in other documents. This approach ends up boosting the importance of words that are unevenly distributed within positive and negative classes while lowering the importance of words that are evenly distributed between positive and negative [12]. As such, Delta TF-IDF allows the word importance to represent better which words are essential to determine if a text is positive or negative. This feature is validated by the results of [12] and the extensive amounts of sentiment analysis research done with Delta TF-IDF yielding great results such as [2]. However, research into using Delta TF-IDF across multiple languages and African languages is specifically limited and merits further study. The following section details the Datasets, Pre-Processing, Feature Extraction, Modeling, and Evaluation utilized for this work.

2 Methodology

2.1 Datasets

We utilized the datasets from AfriSenti-SemEval [15, 20] for Task A (Per-Language Modeling) and Task B (Multilingual Modeling), this totaled to 13 datasets. The 13 datasets comprised of comprising Hausa(HA), Yoruba(YO), Igbo(IG), Nigerian Pidgin (PCM), Amharic(AM), Algerian Arabic(DZ), Moroccan Arabic/Darija(MA), Swahili(SW), Kinyarwanda(KR), Twi(TWI), Mozambican Portuguese(PT), Xitsonga(Mozambique Dialect) (TS), and a combination of all the 12 language datasets for a multilingual task(ALL). With the sourcing of all the datasets coming from Twitter, it allows us to claim that the data reflects the real-world setting. Since there were technical errors in the data release at the time of writing, we do not report our results on Moroccan Arabic/Darija(MA). We use the same splits as Aryal, Prioleau, and Aryal to ensure a fair comparison.

The dataset is roughly balanced by sentiment labels outside of PCM (Nigerian Pigdin), DZ (Algerian Arabic), SW (Swahili), TWI, PT(Mozambican Portuguese), and TS(Xitsonga). For multilingual(ALL), the dataset is balanced by the sentiment labels overall.

2.2 Pre Processing

Data pre-processing was standardized between all approaches tested to ensure result comparability. With the dataset derived from tweets, we removed informal language, emojis, and web links to the best of our ability. Finally, we also removed all known punctuations and English stop words. The pre-processed data was used further for feature extraction.

2.3 Feature Extraction

The pre-processed was passed through model-specific tokenizers for transformer-derived architectures [19]. We utilized a maximum token length of 128 since we found it sufficient to accommodate the longest text across all datasets. Shorter sentences were padded with zeroes. For models that do not rely on tokenization, we extracted Delta TF-IDF features for each language (for language-specific models) and across all languages (for multilingual models). The vectorization was fit on only the train set, while the fit transformation was used on all train, validation, and test sets. The tokenization output was used to train transformer-derived models, whereas standard machine-learning models were trained on Delta TF-IDF features.

2.4 Modeling Paradigm

2.4.1 Feature-based Modeling

Although some languages share similar origins and roots, modern languages are distinct. To evaluate if a dedicated model better supported the uniqueness of each language, we opted to train and tune four operationally different models using Delta TF-IDF features across 11 languages plus the multilingual set. First, we utilize a relatively newer boosted tree-based model, Light Gradient Boosted Machine (lightGBM) [11]. We also used a standard distance-based k-nearest neighbors (kNN) model, a bagging-based Random Forest (RF) model, and a Kernel-based Support Vector Machine (SVM) model. These models were chosen due to their well-established recurring applications within sentiment analysis [14]. The data was trained only on the training split, and the hyperparameters were optimized with 10-fold cross-validation over 40 trials using Bayesian Optimization.

2.4.2 Transformer Modeling

Our modeling approach for transformers followed that of Aryal, Prioleau, and Aryal to enable direct comparison. The significant difference in our approach is selecting the maximum token length utilized; we selected the maximum token length using the longest text (128 tokens), whereas the baseline utilizes a mean text length of 20. The process is repeated for language-specific and multilingual models.

2.5 Evaluation

Evaluation for Per-Language and Multilingual modeling was done on the train and test set. The validation set was not utilized for comparison since the transformer models use it for fine-tuning, whereas the other models do not. We only report weighted F1 scores for language-specific models. Of the scores reported, transformer models only report mean scores, whereas other models report 95% confidence intervals from a 10-fold cross-validation. In contrast, for multilingual models, all standard weighted average classification metrics (F1, Precision, Recall, and Accuracy) are reported in section 3. Finally, we use precision-recall curves to compare the performance of each class for the multilingual model. For the language identification task, we report a table of classification metrics (F1, Precision, Recall, and Accuracy) for each language.

Model training and inference were performed on a late 2021 Lambda Tensorbook with 16 GB Nvidia GeForce 3080. Our code is open source to ensure reproducibility of our work (hidden for anonymity)

3 Results

3.1 Language-Specific Modeling

	Train				Test			
	SVM	LGBM	RF	KNN	SVM	LGBM	RF	KNN
HA	**.97 ± 0.0**	.83 ± 0.0	.97 ± 0.0	.74 ± 0.0	**.75 ± 0.0**	.72 ± 0.0	.73 ± 0.0	.61 ± .01
YO	**.97 ± 0.0**	.88 ± 0.0	.97 ± 0.0	.96 ± 0.0	**.72 ± 0.0**	.67 ± 0.0	.68 ± .01	.61 ± 0.0
IG	.97 ± 0.0	.85 ± 0.0	**.98 ± 0.0**	.97 ± .01	**.78 ± 0.0**	.75 ± 0.0	.76 ± 0.0	.67 ± .01
PCM	.97 ± 0.0	.95 ± 0.0	**.97 ± 0.0**	.97 ± .01	.71 ± 0.0	.68 ± .01	**.72 ± .01**	.67 ± .02
AM	.94 ± 0.0	.68 ± 0.0	**.95 ± 0.0**	.95 ± 0.0	.51 ± .01	.51 ± 0.0	**.54 ± .01**	.51 ± .01
DZ	**.96 ± 0.0**	.70 ± .01	.68 ± .01	.95 ± .01	.61 ± .01	.59 ± .01	**.63 ± .02**	.51 ± .02
SW	**.93 ± 0.0**	.66 ± 0.0	.76 ± 0.0	.58 ± .02	**.53 ± .01**	.51 ± .01	.53 ± .01	.49 ± .03
KR	.95 ± 0.0	.65 ± 0.0	**.95 ± 0.0**	.94 ± 0.0	.54 ± .01	.51 ± .01	**.54 ± .01**	.43 ± .02
TWI	.93 ± 0.0	.85 ± 0.0	**.95 ± 0.0**	.94 ± 0.0	**.63 ± .01**	.55 ± .01	.61 ± .01	.60 ± .03
PT	.87 ± 0.0	.82 ± 0.0	.83 ± 0.0	**.95 ± .01**	.59 ± 0.1	.56 ± .01	**.63 ± .01**	.50 ± .02
TS	**.96 ± 0.0**	.65 ± 0.0	.95 ± 0.0	.95 ± .01	.54 ± .01	.47 ± .02	**.55 ± .02**	.45 ± .04

Table 1: 95% CI of Weighted F1 of Language-Specific Delta TFIDF Models on the Train and Test

	Train			Test		
	AfriBERTa	AfroXLMR	AfroLM	AfriBERTa	AfroXLMR	AfroLM
HA	.83	**.87**	.86	.77	**.78**	.77
YO	**.88**	.79	.78	**.73**	.72	.66
IG	**.90**	.86	.86	**.79**	.76	**.77**
PCM	.72	**.81**	.72	.69	**.74**	.68
AM	.65	.64	**.69**	.58	**.62**	.58
DZ	.41	**.75**	.43	.41	**.65**	.47
SW	.79	**.68**	.68	.62	**.62**	.55
KR	**.79**	.71	.78	.63	**.66**	.56
TWI	**.71**	.61	.61	.56	**.60**	.53
PT	.71	**.76**	.66	.57	**.66**	.44
TS	**.57**	.39	.50	**.44**	.38	.36

Table 2: Mean Weighted F1 of Each Transformer Model on the Test set

When comparing Tables 1 and 2, we notice that while the Language-Specific Delta TF-IDF Models are competitive, the Transformer Models, on average, perform better. However, upon further exploration, when considering the pre-training size and language for transformers, as seen in **outpaper** Data Analysis, also report similar findings. With the lack of similar pre-training context on some languages, models trained on Delta TF-IDF features perform competitively and even outperform the Transformer models. These results demonstrate promising results for our Delta TF-IDF approach on the low-resource languages since they require a significantly lower volume of data than the Transformer Models for similar performance.

As evident from the significantly higher F-1 scores on the train than the test, both approaches overfit. However, the Delta TF-IDF models suffer excessively compared to the Transformer approaches. While the transformers again benefit from finetuning on a held-out validation set during finetuning, the feature-based models did not utilize the data from the validation. They may improve performance further if utilized appropriately. These results support the necessity of further research into African Languages since the general efforts may not work as expected, and specialization may be needed. Finally, since model performance is model-dependent, combined language-specific models for the multilingual modeling task are feasible.

3.2 Multilingual Modeling

	Train				Test			
Models	recall	precision	accuracy	f1	recall	precision	accuracy	f1
AfriBERTa	.85	.85	.85	.85	.68	.69	.68	.68
AfroXLMR	.83	.83	.83	.83	**.69**	**.70**	**.69**	**.69**
AfroLM	.79	.79	.79	.79	.64	.65	.64	.64
SVM	.93	.93	.93	.93	.63	.65	.63	.63
LGBM	.73	.73	.73	.73	.61	.63	.61	.60
RF	**1.0**	**1.0**	**1.0**	**1.0**	.61	.63	.61	.60
KNN	**1.0**	**1.0**	**1.0**	**1.0**	.57	.57	.57	.57

Table 3: Transformers and Delta TF-IDF metrics for multilingual, All the metrics were weighted

Our proposed Delta TF-IDF approach when applied to the multilingual maintains its competitiveness to its transformers counterparts. The performance become more impressive when also accounting for the amount of Data AfroXLMR the highest performing model was trained on to achieve 6% more performance that SVM which was the highest performing Delta TF-IDF model.

4 Conclusion

With the recent release of the AfriSenti-SemEval shared Task 12, hosted as a part of The 17th International Workshop on Semantic Evaluation, 14 new datasets annotated for sentiment analysis on African Languages were made available. We evaluated an approach to this task by utilizing Delta TF-IDF. The approaches were compared to the benchmark set by the current state-of-art transformer models on this task: AfriBERTa, AfroXLMR, and AfroLM. Delta TF-IDF showcased its ability to perform competitively against transformer models with a significant amount of less data. While the field of Sentiment Analysis on African languages is still in its early stages, we hope our work will aid in advancing the field and adding to its literature. To get the field of research into African Languages to where it needs to be, researchers need to allocate the resources, care, and attention needed. We also implore all future researchers and readers to go through our limitations and future below.

5 Limitations & Future Work

While sentiment analysis in NLP is valuable, it also presents risks of abuse, such as surveillance and restriction of freedom. Responsible use of these technologies is crucial to avoid incidents like the Cambridge Analytica scandal. Africa, being in a phase of rapid development, can learn from past mistakes and benefit from research in this field. However, our work has limitations, including the need for more data to improve performance. Establishing partnerships with native speakers and increasing data access from African academic institutions can help address this issue. The use of pre-existing models introduces potential bias and performance issues, and adapting them to African language tokens could be explored in the future. This is seen within the approaches of [5, 6, 3] Computing resources and data cleaning processes may also pose challenges, leading to the loss of valuable information. Language-specific models require more resources with each additional language, while multilingual models may be a promising direction for capturing interdependencies between languages. Lastly, as non-experts in the languages studied, conducting further qualitative analysis is challenging.

References

[1] Jesujoba O. Alabi et al. "Adapting Pre-trained Language Models to African Languages via Multilingual Adaptive Fine-Tuning". In: *Proceedings of the 29th International Conference on Computational Linguistics.* Gyeongju, Republic of Korea: International Committee on Computational Linguistics, Oct. 2022, pp. 4336–4349. URL: `https://aclanthology.org/2022.coling-1.382`.

[2] Muhammad Alkaff, Andreyan Rizky Baskara, and Yohanes Hendro Wicaksono. "Sentiment Analysis of Indonesian Movie Trailer on YouTube Using Delta TF-IDF and SVM". In: *2020 Fifth International Conference on Informatics and Computing (ICIC).* IEEE. 2020, pp. 1–5.

[3] Saurav K Aryal and Howard Prioleau. "Howard University Computer Science at SemEval-2023 Task 12: A 2-Step System Design for Multilingual Sentiment Classification with Language Identification". In: *In Proceedings of the 17th International Workshop on Semantic Evaluation.* 2023.

[4] Saurav K Aryal, Howard Prioleau, and Surakshya Aryal. "Sentiment Analysis Across Multiple African Languages: A Current Benchmark". In: *SIAIA @ AAAI.* 2023.

[5] Saurav Keshari Aryal and Gaurav Adhikari. *Evaluating Impact of Emoticons and Pre-processing on Sentiment Classification of Translated African Tweets.* 2023.

[6] Saurav Keshari Aryal, Hrishav Sapkota, and Howard Prioleau. *Zero-Shot Classification Reveals Potential Positive Sentiment Bias in African Languages Translations.* 2023.

[7] Paul Collier and Jan Willem Gunning. "Why Has Africa Grown Slowly?" In: *The Journal of Economic Perspectives* 13.3 (1999), pp. 3–22. ISSN: 08953309. URL: `http://www.jstor.org/stable/2646982` (visited on 01/21/2023).

[8] Alexis Conneau et al. "Unsupervised Cross-lingual Representation Learning at Scale". In: *CoRR* abs/1911.02116 (2019). arXiv: `1911.02116`. URL: `http://arxiv.org/abs/1911.02116`.

[9] Bonaventure FP Dossou et al. "AfroLM: A Self-Active Learning-based Multilingual Pretrained Language Model for 23 African Languages". In: *arXiv preprint arXiv:2211.03263* (2022).

[10] Jacques Gaillard and Johann Mouton. "The state of science, technology and innovation in Africa: trends, progress and limitations". In: *Science, Technology and Society* (2022), p. 09717218221078548.

[11] Guolin Ke et al. "Lightgbm: A highly efficient gradient boosting decision tree". In: *Advances in neural information processing systems* 30 (2017).

[12] Justin Martineau and Tim Finin. "Delta tfidf: An improved feature space for sentiment analysis". In: *Proceedings of the International AAAI Conference on Web and Social Media*. Vol. 3. 1. 2009, pp. 258–261.

[13] Laura Martinus and Jade Z. Abbott. *A Focus on Neural Machine Translation for African Languages*. 2019. DOI: `10.48550/ARXIV.1906.05685`. URL: `https://arxiv.org/abs/1906.05685`.

[14] Walaa Medhat, Ahmed Hassan, and Hoda Korashy. "Sentiment analysis algorithms and applications: A survey". In: *Ain Shams engineering journal* 5.4 (2014), pp. 1093–1113.

[15] Shamsuddeen Hassan Muhammad et al. "NaijaSenti: A Nigerian Twitter Sentiment Corpus for Multilingual Sentiment Analysis". In: *Proceedings of the 13th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, June 2022, pp. 590–602. URL: `https://aclanthology.org/2022.lrec-1.63`.

[16] Kelechi Ogueji, Yuxin Zhu, and Jimmy Lin. "Small Data? No Problem! Exploring the Viability of Pretrained Multilingual Language Models for Low-resourced Languages". In: *Proceedings of the 1st Workshop on Multilingual Representation Learning*. Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 116–126. DOI: `10.18653/v1/2021.mrl-1.11`. URL: `https://aclanthology.org/2021.mrl-1.11`.

[17] Gorm Rye Olsen. "Western Europe's relations with Africa since the end of the Cold War". In: *The Journal of Modern African Studies* 35.2 (1997), pp. 299–319.

[18] Srinivasan Ramani. "The internet and education in the developing world-hopes and reality". In: *Smart Learning Environments* 2.1 (2015), pp. 1–16.

[19] Thomas Wolf et al. "Huggingface's transformers: State-of-the-art natural language processing". In: *arXiv preprint arXiv:1910.03771* (2019).

[20] Seid Muhie Yimam et al. "Exploring Amharic Sentiment Analysis from Social Media Texts: Building Annotation Tools and Classification Models". In: *Proceedings of the 28th International Conference on Computational Linguistics*. Barcelona, Spain (Online): International Committee on Computational Linguistics, Dec. 2020, pp. 1048–1060. DOI: `10.18653/v1/2020.coling-main.91`. URL: `https://aclanthology.org/2020.coling-main.91`.

Baselining Performance for Multilingual Codeswitching Sentiment Classification*

Saurav K. Aryal, Howard Prioleau
Surakshya Aryal, Gloria Washington
Department of Computer Science
Howard University
Washington, DC 20059
saurav.aryal@howard.edu

Abstract

Sentiment analysis is an essential task in understanding human-generated textual documents. While most research into sentiment analysis focuses on monolingual sentences, in multilingual communities, a significant proportion of social media text contains a mixture of languages or code-switching. Thus, it has become vital to research and build models that handle code-switched data. However, despite significant research and custom expert neural architectures proposed, the current literature is mainly limited to modeling single language pairs. To expand on existing work and baseline performance for this particular task, we perform multiple experiments: fine-tuning pre-trained multilingual models and fine-tuning monolingual BERT models on sentence and word-level translations. The experiments are performed across five datasets where English is code-switched with Spanish, Tamil, Telugu, Hindi, and Malayalam. Our best model outperforms the current best single model that works with multiple code-switched language pairs on standard classification metrics on a binary sentiment classification task. We further expand our experiment with a ternary sentiment classification task and produce results comparable to single language-pair-specific models.

1 Introduction

With the advancements of the internet, communication, and social media technology, the average human generates vasts amount of data. Where a significant portion of this data is human-generated textual data and documents. Significant research has gone into sentiment analysis to automate our understanding of unstructured textual data. However, most research focuses on utilizing monolingual documents across multiple languages. In contrast, estimates suggest that about half of the world population is bilingual and such users routinely produced textual data resulting from a mixture of two or more languages.

Linguistic code-switching is the concurrent use of two or more languages or dialects in a conversation. Code-switching, in general, has been studied extensively in psycho-linguistics and sociolinguistics [14, 13, 16]. Since the linguistic definition of code-switching is multifaceted, this paper, for readability, will refer to only the linguistic variant as code-switching. Sentiment analysis of code-switched data is not only limited but also a more challenging task because the amalgamation of two source languages creates a secondary low-resource language that partially maintains the features of the primary language while creating a distinct topology that models may need to learn separately.

Despite existing linguistic challenges, research into computational tasks related to code-switching has been increasing. However, sentiment analysis, in particular, suffers from a paucity of labeled datasets and language pairs. To further work towards the development of a single model for the sentiment classification of multiple code-switched language pairs, in this work, we review relevant literature on code-switching, broader natural language processing (NLP), and automated pattern recognition. Upon shortlisting feasible approaches, we detail our proposed methods, perform experiments, and present our results as a baseline for future attempts for sentiment classification of code-switched text. The following sections of the paper will discuss our findings, limitations, future work, and conclusions.

2 Relevant Works

This section will present our review of the literature on broader NLP sentiment analysis, code-switching sentiment analysis, and automatic machine learning in separate subsections.

2.1 Code-Switching and Sentiment Analysis

Progress in multilingual models and sentiment spurred the interest of researchers in the code-switching domain. The brunt of the work has been focused on creating corpora [10, 8, 2, 2, 11], language and parts of speech tagging [18, 17], and language modeling. In the following subsections, we narrow our focus to datasets, existing approaches, and translations within the code-switching domain.

2.1.1 Datasets

Datasets for sentiment analysis for codeswitched text are limited, likely due to the requirement of manual labeling. Spanish-English and Hindi-English datasets were provided as part of Task 9 for the International Workshop on Semantic Evaluation (SemEval) 2020 [12]; both language sets are sourced from social media texts. Similarly, Malayalam-English and Tamil-English came from the Dravidian-CodeMix challenge from the Forum for Information Retrieval Evaluation (FIRE) 2020 [4, 5] both sets were sourced from Youtube comments. Lastly, the Code-Mixed Telugu-English Text (CMTET) dataset contains Telugu-English text, which is from a published work, and was also sourced from social media text [9]. Furthermore, as this research domain is novel, all datasets contain real-world challenges such as informal language usage, casual transliterations, and spelling errors. To the best of our knowledge, these are the only standard annotated research datasets currently existing.

2.1.2 Translating Code-Switched Text

The most naive yet intuitive solution to handling sentiment classification of code-switched text might be to apply translation to the text. Then, we can fine-tune established pre-trained BERT models for a downstream sentiment classification task. While the approach has its merits, there are caveats with translating code-switched text. Similar to sentiment analysis, work in code-mixed translation is limited since most of the work for machine translation has centered around improving monolingual translations and low-resource language problems. However, a recent work that provided a Hindi-English codeswitching translation dataset and translation tests suggested that openly available translation sources such as Google Translate may not directly perform well on code-switched sentences [19]. Furthermore, additional challenges persist, including but not limited to transliteration, word-level language ambiguity, spelling variations, named entity recognition, informality, substandard punctuation and grammar, and missing context. Regardless of the complexity of accurate translation, the proposed sentiment analysis task utilizing sentence-level and word-level translations are worth testing since accurate translation, and the carrying over of sentiment may not be mutually exclusive.

3 Methodology

Multiple experiments are performed to progress toward a single model that can accurately identify the sentiment of multiple code-switch language pairs. This section details the datasets, the experiments performed, the pre-processing approaches employed, and the modeling and evaluation approach used in distinct subsections.

3.1 Datasets

We utilized all five datasets available in the current research literature. The dataset includes Spanish-English and Hindi-English [12], Tamil-English and Malayalam-English [4, 5], and Telugu-English [9]. The similarity of the sourcing of each dataset allows us to claim performance mirroring the real world. Furthermore, to our knowledge, this is the first work that experimented with all five datasets. Additionally, each dataset has differing sample sizes; we can find the splits utilized for the experiments in Table 1 and the frequency of each class label in Table 1.

Language	Neu	Pos	Neg	Total	Train	Val	Test
Hindi-English	7,492	6,616	5,892	20,000	14,000	3,000	3,000
Spanish-English	2,593	8,922	4397	15,912	12,194	1,859	1,859
Tamil-English	1,801	10,559	2,037	14,397	10,358	1,163	2,876
Malyalam-English	403	2,811	738	3,952	2,860	319	773
Telugu-English	4,222	7,929	7,717	19,868	11,920	3,974	3,974

Table 1: Sentence Labels and Sample Sizes By Language Pair

3.2 Experiments

Based on our extensive literature review, we perform the following sets of experiments with the datasets:

3.2.1 Experiment 1: Translations

Translations into English provide access to high-performing BERT-based models for sentiment classification. Although the translation may not be accurate on code-switched text [19], four types of translations are performed: they are detailed in the section 3.3 below. No hyperparameter optimization was performed.

3.2.2 Experiment 2: Fine-tuning Multilingual Models

Massive multilingual models have been trained to learn the representation of multiple languages [1, 15]. They have been successfully fine-tuned for downstream sentiment classification of text from multiple languages, but training data for the models are monolingual. We experiment with fine-tuning the pre-trained sentiment classification model across five different language pairs. No hyperparameter optimization was performed.

The above experiments were performed twice each, first for a binary classification problem (positive vs. negative) and again for a ternary classification problem (positive, negative, and neutral). The following sections will refer to these experiments and detail any relevant pre-processing and modeling.

3.3 Pre-processing

Pre-processing can be divided into two major types based on the proposed experiments: approaches that utilized translations and those that did not.

3.3.1 Translation Approaches

In Experiment 1, we employed the Google Translation API to translate code-switched datasets into English. To ensure accuracy, we performed transliteration from roman-text to the original Indic languages using an open-source library[3]. Recognizing that direct translations may not be precise [19], we conducted translations at both the sentence and word levels. Additionally, we considered scenarios where the source language was known or unknown, toggling the translation API's automatic language detection feature. An example of the translation process involved an English-Spanish code-switched sentence: "he said juntate con nosotros i te aremos rica." The four possible translations were as follows: (1) sentence-level translation with known source language: "he said join us and we will make you rich," (2) sentence-level translation with unknown source language: "he said join us and we will make you rich," (3) word-level translation with known source language: "I have said get together with us Yo tea we will delicious," and (4) word-level translation with unknown source language: "he said board con us i the we will ricas." These translation approaches were employed to address the challenges of code-switching analysis.

3.3.2 Untranslated Approach

For our Untranslated experiments, we clean the data by removing the English stop-words, punctuation, links, and digits from the sentences and denoising social media text. This process makes the models' embeddings much more accurate since that data type tends to interfere with useful sentence information.

After cleaning, the cleaned sentences as passed through the model-specific tokenizer. We set the max sentence token value to 25 by looking at the average number of tokens in a sentence across all datasets. The sentences with fewer than 25 tokens were padded with zeros. For Experiment 2, the data is passed to a pre-trained model to be fine-tuned for classification.

3.4 Modeling and Evaluation

For Experiments 1 and 2, the Fine-tuning models we utilized come from the Hugging Face library [20]. We used three models, which include TweetEval[1] model, Xlm-Roberta-Base[6] model, and distilUSE-base-multilingual-case[15].

We utilize the same experimental protocol and splits for the binary and ternary classification problems for direct comparison with the baseline. However, the binary task is not trained or evaluated on Telugu for direct comparison to the current best-performing model [7]. Model training and inference were performed on a late 2021 Lambda Tensorbook with 16 GB Nvidia GeForce 3080.

To evaluate these experiments, we calculate standard classification metrics of Precision, Recall, Accuracy, and macro F-1 score. The top-performing models for each experiment set are tabulated in section 4. Since the classification task is not balanced, as seen in Table 1.

4 Results

4.1 Binary Classification

Model Source	Precision	Recall	Accuracy	F1
TweetEval[7]	0.84	0.84	0.84	0.84
xlm-roberta-large	**0.85**	**0.85**	**0.85**	**0.85**
distiluse-v1	0.84	0.84	0.84	0.84

Table 2: Experiment 1: Top Performing Finetuned models

The best-performing models across Experiment 1 for Binary sentiment classification are tabulated in Table 2. We notice that word-level translations marginally outperform direct sentence-level translations for this task, although the difference is not significant. One average metrics and across all languages, we see that this method also beats the multilingual baseline model Gupta et al. for the same datasets and task. No significant difference was noticed between experiments where the source language was known or unknown.

Model Source	Translation	Precision	Recall	Accuracy	F1
TweetEval	by Sentence	**0.85**	**0.85**	**0.85**	**0.84**
xlm-roberta-large	by Sentence	0.84	0.85	0.85	0.84
distiluse-v1	by Sentence	0.84	0.84	0.84	0.83
TweetEval	by Word	0.86	**0.86**	**0.86**	**0.85**
xlm-roberta-large	by Word	**0.86**	0.86	0.86	0.85
distiluse-v1	by Word	0.86	0.86	0.86	0.85

Table 3: Experiment 2: Top Performing Translation + Finetuned models

For Experiment 2, multilingual models were fine-tuned on a merger of all four datasets without translations. The average performance metrics of this experiment and the top 3 performing models can be found in Table 2. Average performance across most models is the same despite their difference in pre-training data. The results are also strikingly similar to the top results in Experiment 1, which implies that the translation step may be unnecessary.

4.2 3-Class Results

Previous experiments have been conducted on individual datasets, but no attempts have been made to perform a ternary classification task across all five language pairs. Our work establishes the baseline for this task and dataset combination.

Model Source	Precision	Recall	Accuracy	F1
TweetEval	0.70	0.70	0.70	0.70
xlm-roberta-large	**0.72**	**0.72**	**0.72**	**0.72**
distiluse-v1	0.69	0.69	0.70	0.69

Table 4: Experiment 1: Top Performing Finetuned models

For Experiment 1, the performance metrics for the best-performing models for Ternary sentiment classification are tabulated in Table 5. We notice that word-level translations significantly outperform direct sentence-level translations for this task. While this was not observed in the simpler binary classification task, this task, with the introduction of the challenging neutral class, allows for difficult decision boundaries causing sentential translations to struggle in comparison to word-level counterparts. Furthermore, despite the change in topology and likely incorrect translations, the word-level translations performed much better than a naive random predictor.

Model Source	Translation	Precision	Recall	Accuracy	F1
TweetEval	Sentence	**0.66**	**0.67**	**0.67**	**0.66**
xlm-roberta-large	Sentence	0.23	0.49	0.49	0.32
distiluse-v1	Sentence	0.66	0.67	0.67	**0.66**
TweetEval	Word	**0.70**	**0.70**	**0.70**	**0.70**
xlm-roberta-large	Word	0.24	0.49	0.49	0.32
distiluse-v1	Word	0.69	0.70	0.70	0.69

Table 5: Experiment 2: Top Performing Translation + Finetuned models

For Experiment 2, multilingual models were finetuned on a merger of all five datasets without translations. The average performance metrics of this experiment and the top 3 performing models can be found in Table 2. Unlike the binary task, the average performance across most models is different for the ternary task. The results are also strikingly similar to the top results in Experiment 1, which implies that the translation step may end up hurting performance for more complicated classification tasks, possibly due to the loss of topological information of each language.

5 Conclusion

In summary, we experimented with translation techniques and fine-tuning multilingual models for sentiment classification in code-mixed languages. We found a single model that outperformed the only comparable model for binary classification. Translations and fine-tuning without translation produced similar results for binary classification, but adding a neutral class degraded performance. Untranslated fine-tuning of XLM-based models performed best for the 3-class problem across five languages.To our knowledge, we are the first to attempt the ternary classification task across these 5 disparate datasets. Our model significantly outperforms language-specific models. More work is needed to improve both binary and ternary classification.

Limitations & Future Work

Despite improving the performance metrics of the baseline, our proposed approach still has room for further improvement. Additionally, the lack of comparable work on Dravidian languages and domain-specific dependencies limits direct comparisons. The current datasets only cover code-switching between English and other languages, while other language pairs remain unexplored. The proposed algorithm relies on accurate language representations and assumes the identification of languages, which can be challenging for low-resource languages. Human expertise is also needed to create word sets for positive and negative sentiments in the target language.

Future work requires, increased research funding and larger, more challenging datasets are needed to advance this field. This would require creating datasets beyond the current language pairs is crucial. While the approach is rudimentary and requires refinement, it outperformed the language-specific custom model of the baseline. Future research should consider alternative perspectives beyond novel neural network architectures and address the specific challenges of this task.

References

[1] Francesco Barbieri et al. "TweetEval:Unified Benchmark and Comparative Evaluation for Tweet Classification". In: *Proceedings of Findings of EMNLP*. 2020.

[2] Anab Maulana Barik, Rahmad Mahendra, and Mirna Adriani. "Normalization of Indonesian-English code-mixed Twitter data". In: *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*. 2019, pp. 417–424.

[3] Irshad Ahmad Bhat et al. "Iiit-h system submission for fire2014 shared task on transliterated search". In: *Proceedings of the 6th Annual Meeting of the Forum for Information Retrieval Evaluation*. 2014, pp. 48–53.

[4] Bharathi Raja Chakravarthi et al. "A sentiment analysis dataset for code-mixed Malayalam-English". In: *arXiv preprint arXiv:2006.00210* (2020).

[5] Bharathi Raja Chakravarthi et al. "Corpus creation for sentiment analysis in code-mixed Tamil-English text". In: *arXiv preprint arXiv:2006.00206* (2020).

[6] Alexis Conneau et al. "Unsupervised Cross-lingual Representation Learning at Scale". In: *CoRR* abs/1911.02116 (2019). arXiv: `1911.02116`. URL: `http://arxiv.org/abs/1911.02116`.

[7] Akshat Gupta et al. "Unsupervised self-training for sentiment analysis of code-switched data". In: *arXiv preprint arXiv:2103.14797* (2021).

[8] Simran Khanuja et al. "A new dataset for natural language inference from code-mixed conversations". In: *arXiv preprint arXiv:2004.05051* (2020).

[9] Siva Subrahamanyam Varma Kusampudi, Preetham Sathineni, and Radhika Mamidi. "Sentiment Analysis in Code-Mixed Telugu-English Text with Unsupervised Data Normalization". In: *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*. 2021, pp. 753–760.

[10] Holy Lovenia et al. "ASCEND: A Spontaneous Chinese-English Dataset for Code-switching in Multi-turn Conversation". In: *arXiv preprint arXiv: 2112.06223* (2021).

[11] Dau-Cheng Lyu et al. "Seame: a mandarin-english code-switching speech corpus in south-east asia". In: *Eleventh Annual Conference of the International Speech Communication Association.* 2010.

[12] Parth Patwa et al. "SemEval-2020 Task 9: Overview of Sentiment Analysis of Code-Mixed Tweets". In: *Proceedings of the 14th International Workshop on Semantic Evaluation (SemEval-2020).* Barcelona, Spain: Association for Computational Linguistics, Dec. 2020.

[13] Shana Poplack. *Code Switching: Linguistics. International Encyclopedia of the Social & Behavioral Sciences, ed. Niel Smelser and Paul Baltes.* 2015.

[14] Shana Poplack, David Sankoff, and Christopher Miller. "The social correlates and linguistic processes of lexical borrowing and assimilation". In: (1988).

[15] Nils Reimers and Iryna Gurevych. "Making Monolingual Sentence Embeddings Multilingual using Knowledge Distillation". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing.* Association for Computational Linguistics, Nov. 2020. URL: `https://arxiv.org/abs/2004.09813`.

[16] Gillian Sankoff. *Language use in multilingual societies: some alternative approaches.* Penguin Books, 1972.

[17] Thamar Solorio and Yang Liu. "Learning to predict code-switching points". In: *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing.* 2008, pp. 973–981.

[18] Victor Soto and Julia Hirschberg. "Joint part-of-speech and language ID tagging for code-switched data". In: *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching.* 2018, pp. 1–10.

[19] Vivek Srivastava and Mayank Singh. "PHINC: A parallel Hinglish social media code-mixed corpus for machine translation". In: *arXiv preprint arXiv:2004.09447* (2020).

[20] Thomas Wolf et al. "Transformers: State-of-the-Art Natural Language Processing". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations.* Online: Association for Computational Linguistics, Oct. 2020, pp. 38–45. URL: `https://www.aclweb.org/anthology/2020.emnlp-demos.6`.

Phishing Resistant Systems: A Literature Review*

Student Paper Abstract

Jonathan Luckett
College of Business, Innovation, Leadership and Technology
Marymount University
Arlington, Virginia 22207
jonathan_luckett@marymount.edu

Phishing attacks account for 90 percent of all data breaches. There are a number of solutions to mitigate these attacks. This research provides a broad literature review of phishing-resistant systems including Microsoft solutions, FIDO2 protocols, email authentication standards and protocols, and browser-based detection systems. The reviewed literature is categorized by Fraudulent Websites, Compromised Credentials, Compromised CSP, and Phishing emails. While there are several promising phishing resistant systems in the market today, no single product provides full protection against phishing attacks.

Predictive Modeling for Customer Purchase Behavior: A Logistic Regression Approach Based on Age and Estimated Salary[*]

Student Paper Abstract

Nida Peerzada
Marymount University
Arlington, Virginia 22207
nap78009@marymount.edu

This research paper aims to develop a logistic regression model to predict whether a customer will make a purchase or not based on their age and estimated salary. The project addresses the increasing importance of data-driven decision-making in the business world and aims to assist marketing teams in identifying potential customers who are likely to purchase, thereby optimizing marketing efforts and increasing the product's sales. The dataset used in the project is sourced from a marketing campaign of a particular product and consists of customer information such as age, gender, and estimated salary. The target variable is whether the customer made a purchase or not. The logistic regression model developed in this project can help businesses to improve their marketing strategies, reduce costs, and make data-driven decisions. The paper describes the process of loading and cleaning the data, defining variables, splitting the data, fitting and transforming using Standard Scalar, training and making predictions with the logistic regression model, and evaluating the model's performance using accuracy, precision, recall, and F1 score metrics. The results demonstrate that the model accurately predicts whether a customer will make a purchase based on their age and estimated salary. It can assist marketing teams in identifying potential customers who are likely to purchase, resulting in increased sales and revenue.

Malware Detection with Hybrid Datasets in Machine Learning Model Experiments[*]

Student Paper Abstract

Bipun Thapa and Tarie Lee
Marymount University
Arlington, Virginia 22207
{bst77492, t0177636}@marymount.edu

Machine learning paves the way for computers to predict, draw inferences, or forecast based on modeling. These machine learning training models are highly dependent on large datasets; unfortunately, the required datasets are unavailable. The two leading causes are that the datasets do not exist, and if they do exist, privacy concerns prevent the data's release. One solution is to create synthetic data. This work sets out to answer how good is synthetic data and real data combined, does it produce better results than real data? And do results improve by selecting alternate features, hence more data fields? Since there is no shortage of information pertaining to malware, these experiments use a malware dataset. This work is significant in ascertaining how synthetic data, when combined with real data, affects machine learning models. Since there is no documented work in this area, it is important to note that the experiments could not prove that synthetic data was helpful in machine learning modeling. It is recommended that further investigation is warranted as to why the experiments did not yield better results.

Investigating Government-Funded Cyber Influence Operations in the Social Media Era[*]

Student Paper Abstract

Jacob Stedman
School of Technology and Innovation
Marymount University
Arlington, Virginia 22207
jds55559@marymount.edu

Computers are the new platform for cyber operations by foreign threat actors to influence the climate of political topics. This paper aims to categorize keywords of a small data set disclosed by Reddit's transparency security report. Typically, when social media companies address misinformation on a platform, the posts are removed from the public-facing, and the company's security team deals with it internally. This research aims to take advantage of the transparency within this report and the available data to determine if keywords within threat actors' posts have a reoccurring theme associated with them. The goal is to develop a machine-learning model using Python code to weigh the keywords and, at a future point, test it to determine the likelihood of a specific post being something intended to be malicious information.

Brain Tumor Detection Using Deep Learning Techniques [*]

Student Paper Abstract

Ramya Gora and Anusha Nandru
Sacred Heart University
Fairfield, CT 06825
{gorar, nandrua}@mail.sacredheart.edu

Brain tumor detection is a critical task in the field of medical image analysis. Accurate and timely detection of tumors is essential for proper treatment planning, which in turn can significantly improve patient outcomes. In recent years, the use of deep learning techniques, especially Convolutional Neural Networks (CNN), has shown promising results in detecting brain tumors in medical images. This paper presents a generalized abstract for brain tumor detection using CNN. Creating machines that behave and work in a way like humans is the objective of artificial intelligence (AI). In addition to pattern recognition, planning, and problem-solving, computer activities with artificial intelligence include other activities. A group of algorithms called "deep learning" is used in machine learning. With the aid of magnetic resonance imaging (MRI), deep learning is utilized to create models for the detection and categorization of brain tumors. This allows for the quick and simple identification of brain tumors. Brain disorders are mostly the result of aberrant brain cell proliferation, which can harm the structure of the brain and ultimately result in malignant brain cancer. The early identification of brain tumors and the subsequent appropriate treatment may lower the death rate. In this study, we suggest a convolutional neural network (CNN) architecture for the efficient identification of brain tumors using MR images. This paper also discusses various models such as ResNet-50, EDA, CNN, Train and Evaluate Model and conducts a comparison between the proposed architecture and these models.

[*] Copyright is held by the author/owner.

Integrating Cybersecurity and Aerospace Manufacturing Quality with the Aid of Zero Knowledge Proof (ZKP) to Provide a Secure and Confidential Way to Transfer Classified Aerospace Data*

Student Paper Abstract

Aswin Krishna Balachandran Pincy
Department of Computer Science
Bay Atlantic University
Washington, DC 20005
abalachandranpincy@stu.bau.edu

This research paper investigates the potential for zero-knowledge proofs (ZKP) to improve the quality control of aerospace components. Modern manufacturing techniques and evolving cybersecurity threats present obstacles for traditional quality control methods. ZKP is a cryptographic technique that can verify the quality of components and the integrity of manufacturing processes without divulging sensitive information. This research paper investigates the potential applications, advantages, and obstacles associated with ZKP implementation in aerospace manufacturing. ZKP appears to be a promising method for enhancing the security, privacy, and dependability of aerospace systems, however, additional research is required to address its limitations and practical considerations. This research paper emphasizes the significance of continuous innovation in aerospace manufacturing quality control to ensure safety, dependability, and quality in an environment that is swiftly changing.

Reviewers — 2023 CCSC Eastern Conference

Alvin, Chris	Furman University, Greenville, SC
Anewalt, Karen	University of Mary Washington, Fredericksburg, VA
Braught, Grant	Dickinson College, Carlisle, PA
Carter, Karla	Bellevue University, Bellevue, NE
Childs, Dawn	Marymount University, Cypress, CA
Conrad, Sue	Marymount University, Fairfax, VA
Costa, Mónica Isabel Teixeira	Polytechnic Institute of Castelo Branco, Castelo Branco, Portugal
D'Antonio, Lawrence	Ramapo College, Dobbs Ferry, NY
Dimitoglou, George	Hood College, Frederick, MD
DiTursi, Dan	Siena College, Loudonville, NY
Dougherty, John	Haverford College, Haverford, PA
Finlayson, Ian	The University of Mary Washington, Fredericksburg, VA
Flinn, Michael	Frostburg State University, Frostburg, MD
Freedman, Reva	Northern Illinois University, DeKalb, IL
Gaspar, Alessio	University of South Florida, Tampa, FL
Green, Nathan	Marymount University, Reston, VA
Grinberg, Grigoriy	Montgomery College, Gaithersburg, MD
Gupta, Pranshu	DeSales University, Center Valley, PA
Highley, Timothy	La Salle University, Philadelphia, PA
Hovemeyer, David	Johns Hopkins University, Baltimore, MD
Hu, Jenny	Valencia College, lakeland, FL
Lee, Ingyu	Troy University, Troy, AL
Lindoo, Edward	CCSC Treasurer, Stuart, FL
Lopez, Christian	Lafyaette College, Easton, PA
McCloskey, Robert	University of Scranton, Scranton, PA
Ngo, Linh	West Chester University of Pennsylvania, West Chester, PA
Nguyen, Hoang Huu	University of Illinois at Chicago, Saint Paul, MN
Poger, Sofya	SP Software Company, Woodland Park, NJ
Ravishankar, Veena	University of Mary Washington, Fredericksburg, VA
Rizvi, Syed	Pennsylvania State University, Altoona, PA
Rosiene, Carolyn Pe	University of Hartford, West Hartford, CT
Ryan, Kathleen	DeSales University, Center Valley, PA
Sanders, George	SUNY-Buffalo, Amherst, NY
Sanders, Patrick	SUNY University at Buffalo, Buffalo, NY
Senbel, Samah Ahmed	Sacred Heart University, Fairfield, CT
Stange, Melissa	Laurel Ridge Community College, Middletown, VA
Stedman, Jacob Daniel	Marymount Univ./Advanced Circuits, Osseo, MN

Teresco, James D	Siena College, Loudonville, NY
Tu, Junyi	Salisbury University, Salisbury, MD
Webb, Alla	Montgomery College, Gaithersburgh, MD
Wonnacott, David Glenn	Haverford College, Haverford, PA
Wright, John	Juniata College, Huntingdon, PA
Xu, Wenjuan	Frostburg State University, Frostburg, MD
Zavala, Laura	Medgar Evers College, CUNY, Brooklyn, NJ