North Andover, MA 01845 | **(978) 304-9373** | erichsu137@hotmail.com |

Eric Hsu

PROFESSIONAL SUMMARY

An M.S. graduate specializing in applied AI and data science, with a robust interdisciplinary research background, including significant projects in psychology and neuroscience. Demonstrated expertise in leveraging AI technologies and data analytics to solve complex problems across various disciplines. Skilled at working independently and collaboratively, self-driven to learn, utilize new technologies, and share knowledge with team members in multiple disciplines to improve team productivity. Passionate about leveraging software development and data analytics to drive innovation and contribute to meaningful and humanitarian projects across various fields, including healthcare, neuroscience, and psychological research.

Seeking introductory work opportunities within the software development and data science domain to gain hands-on experience and further develop skills. Eager to contribute to impactful projects, learn from experienced professionals, and make a positive impact in the field.

Education

University of Massachusetts, Amherst — M.S. Computer Science

Degree obtained: 2023 | Amherst, MA

I graduated with a Master of Science in computer science in 2023. While attending full-time at UMASS Amherst, I assisted SEED Lab run by Professor Adam Grabell on their Earlyscreen project led by Ph.D. candidate Manasa Kalanadhabhatta, focused on the use of artificial intelligence to classify early signs of childhood psychopathology from using facial feature data collected during frustration tasks. For the summer and fall of 2023, I assisted Professor Ravi Karkar and Ph.D. candidate Abhay Anand on their Microtasks & DCT Clock Test project, which aimed to find a better testing method for dementia progression.

University of Massachusetts, Amherst — B.S. Computer Science

Degree obtained: 2020 | Amherst, MA

I graduated with a Bachelor of Science in computer science in 2020. While completing credits required for the major degree, I completed coursework to gain a minor in mathematics.

Experience

Job title: Game Development Intern / Research Assistant
Employer: Self-Regulation, Emotions, and Early Development Lab (SEED Lab)

(6/1/2022 – Present) | Full-time (6/1/2022 – 9/1/2022), Part-time (present)

Summary of position and duties

I have worked for SEED Lab, a research laboratory that is part of the psychology department at UMASS Amherst focused on studying emotional regulation in young children, for nearly 2 years under Professor Adam Grabell, and advised by Ph.D. candidate Manasa Kalanadhabhatta. SEED Lab seeks to contribute to a better understanding of how young children learn to understand and control their feelings, and how to better identify and treat children at risk for mood and behavior problems. Because much of children's emotional regulation is shaped by their caregivers and environment, SEED Lab's participants come from a diverse set of racial and ethnic backgrounds. SEED Lab's method for conducting studies involves developing and studying games that are exciting, fun, and teach children about their feelings.

During my time as a Game Development Intern, I was responsible for the recreation of their laboratory frustration game, Amazing Cake Game, within the Unity Game Engine as a stand-alone application, implementing the data pipeline from the application to the university's secure servers, handling the distribution of the game application, and troubleshooting any issues that participants encountered as part of their Earlyscreen project. As a research assistant, I transitioned to working on documenting the feature extraction process for psychology research assistants to recreate as I worked on the hyperparameter tuning and training of the classification models.

Responsibilities

- Game Development: Recreated the Amazing Cake Game in Unity 2D with sound effects, scene transitions, and interactive UI/UX. Implemented series of data collection behaviors within the game's programming logic that include timestamp/event recording, face cam recording, encrypted data chunk sending, and microphone recording. Documented how to distribute the game to participants through Apple's TestFlight program for use by other research assistants. Built Windows version of game application alongside own installer for the distribution method using InnoSetup.
- Technical Support: Troubleshooted technical issues encountered by participants and brainstormed solutions encountered by research assistants on how to differentiate data received by multiple participants.
- Data Pipeline and Data Safety: Leveraged Unity's support for TLS 1.2 for SSL encryption, programmed the game to send encrypted data chunks and delete participant files from participant devices afterwards. Implemented decryption on server-side Flask application to reformulate participant data files from encrypted chunks.
- Data Processing: Investigated and documented how to use Openface to perform feature extraction on participant video files and walked other research assistants on how to use said software. Used the Pandas library in python to further process the data output by Openface into the right format for use in classification models.
- Classification Model Development: Managed the training, hyperparameter tuning, and evaluation of the single-instance-learning and fusion model pipelines using the SKLearn and Pandas libraries. Hyperparameter tuning includes testing classifiers, parameters, feature vectors, and models.
- Collaboration and Communication: Communicated highly technical terminology and concepts to individuals with varying levels of expertise, including those outside of computer science, ensuring clear understanding and effective collaboration. Such examples include game logic, data processing pipelines, and model functionality.

Related skills

Unity game engine (C#); Python (SKLearn, Pandas, Flask); Docker; Linux/Unix terminal commands; Application development; Application deployment; UI/UX; Data science; Data processing; Model training/tuning; Documentation; Communicate technical terms to non-technical individuals; Ability to work with other disciplines; Ability to work independently; Self-starter; Driven to learn.

Job title: iOS Developer / Research Assistant
Employer: University of Massachusetts Amherst

(7/1/2023 – 12/1/2023) | Part-time

Summary of position and duties

My responsibilities as an iOS developer for Professor Ravi Karkar and Ph.D. candidate Abhay Anand involved developing a drawing application for the iPad for use in the data collection process of their Microtasks & DCT Clock Drawing Test project and assisting in the formulation of the data processing procedure before participant data collection began. My duties also include testing and implementing any features requested by the professor and leading Ph.D. student, with a focus on optimizing UI/UX for people with progressing levels of dementia or who are relatively older.

Responsibilities

- **Application Development:** Developed iPad drawing application using SwiftUI and the PencilKit library. Data collection features include saving Apple Pencil's metadata, participant drawn images, and timestamps taken to finish each task within the application and sending data to Google Drive via sharing functionality in Swift.
- **Documentation:** Documented feature requests made by leading professor and Ph.D. student on the projects as well as keeping update logs via Git repository. To make knowledge transferable, documented how to run and build the application onto the iPad device.
- **UI/UX considerations:** Implemented functionality to start and stop the test to give participants the ability to take a break mid-test. UI interactive elements are strategic places to avoid unintended interactions from participant's forearm. To accommodate expected participant disabilities, implemented measures to prevented rapid button presses caused by shaky movements.
- **Data Processing Procedure Planning:** Investigated ways of feature extraction and abstraction for collectible data from the drawing application to emphasize feature indicators found to be significant from prior neuroscience studies. Options explored include using built-in image similarity scoring based on ground-truth drawings made by multiple individuals defined as cognitively healthy and taking the average of all comparisons as the scoring metric. Dynamic Time Warping was considered to negate the effect on how much time a drawing took for individuals. Analyzed the time taken to think about given tasks to study cognitive function and its correlation to disease progression.
- **Testing and Quality Assurance:** Professor and Ph.D. student simulated as participants taking the test, looking for any possible edge case. Made changes to application in response to any unexpected behavior.
- **Collaboration and Communication:** Met the leading Ph.D. student 1-2 times a week to discuss any progress made during the past week, plans for the next week and the timeline for the study.

Related skills

SwiftUI (PencilKit); Python (OpenCV); Version control (Git); Application development; Application deployment; UI/UX; Data science; Data abstraction; Data processing; Quality Assurance; Documentation; Ability to work independently; Self-starter; Driven to learn.

Job title: Grader for CS365: Digital Forensics
Employer: University of Massachusetts Amherst

(9/1/2022 – 12/1/2022) | Part-time

Summary of position and duties

My responsibilities as a grader for CS365: Digital Forensics course taught during the Fall 2022 semester included coordinating with the teaching assistant, other graders, and undergraduate course assistants to grade assignments and handle regrade requests in a timely and fair manner using the Gradescope grading system.

Responsibilities

- **Collaboration and Communication:** Coordinated with the course TA, UCAs, and graders on which questions and assignments each person is responsible for grading and handling regrade requests for. Updated everyone on finished grading assessments and verifying grading with TA and professor.
- **Rubric Design:** Compartmentalized key points required for each answer within Gradescope's grader user interface and assigned significance accordingly to be most fair to the subject material.
- **Grading:** Accurately evaluated student assignments according to established rubrics and guidelines. Providing constructive feedback to students to support their learning and development.
- **Sustain Expertise:** Continuously updated and expanded on subject expertise in digital forensics to ensure accurate and effective assessment practices.

Related skills

Attention to detail; Good communication; Ethical conduct; Time management; Collaboration; Continuous Learning; Ability to work independently; Self-starter.

Job title: Student Intern Developer
Employer: MyConnectUSA

(6/1/2021 – 11/1/2021) | Part-time

Summary of position and duties

MyConnectUSA is a startup focused on developing a social media platform geared towards families and caretakers of those with disabilities. Other social media platforms, like Facebook, are the primary resource for families looking into treatment options and connecting with others. MyConnectUSA seeks to go a step further and provide the same services but based on their geographic location, which is a prevalent problem on other social media platforms.

As a Student Intern Developer, I collaborated with a multidisciplinary team of developers, designers, and data collectors to build both the frontend and backend functionalities of various projects using the Bubble.io low-code/no-code platform. My role involved translating Figma designs into functional and visually appealing interfaces, as well as implementing new feature requests from founders, project managers, and testers.

Responsibilities

- **Website Development:** Developed frontend UI following designer Figma designs that gets populated with data from the bubble.io backend database tables. Built the following pages on the website:
  - **Emailing system:** Base emailing system with inbox and message writing system that allows for searching of other users. Implemented blocking system to prevent messages from being sent by blocked users.
  - **Newsfeed:** General newsfeed populated with posts based on the user's preferences and connections. Supports reply and re-reply functionality, as well as liking and sharing. Allows for users to make posts and choose whether level of privacy for posts (i.e., public, connections-only)
  - **Settings page:** General settings for the user account.

    - **Article page:** Base template for article page that is auto filled based on the pulled data from the backend that will be auto formatted into the page. Includes an article input page to write and format new articles.
- **Collaboration and Communication:** Met with project managers and other designers to collaboratively improve productivity and assist other developers with using the bubble.io platform to develop necessary functionality for other pages. Regularly participated in stand-up meetings to discuss progress, volunteer for undeveloped pages, and discuss plans for the day.
- **Quality Assurance:** Personally tested functionality to check for bugs. Stress tested by project managers and testers to verify any issues in functionality or non-user-friendly UI/UX choices.

Related skills

Figma; bubble.io; Website development; UI/UX; Database design; Standup meetings; Agile Methodology; Effective knowledge transfer; Good communication; Collaboration; Ability to work with other disciplines; Ability to work independently; Self-starter; Driven to learn.

Job title: Technical Writer
Employer: AIBT

(5/1/2020 – 7/1/2020) | Part-time

Summary of position and duties

As a contract Technical Writer for AIBT, my primary responsibility is to ensure the accuracy and clarity of the documentation related to the functionality and configuration of the ion implanter. This involved proofreading, updating existing documents, and creating new ones as needed. My role is crucial in providing clear, concise, and accurate information to users and stakeholders, facilitating their understanding and effective use of the ion implanter.

Responsibilities

- **Proofreading and Editing:** Ensured technical documents were free of grammatical, spelling, and punctuation errors. Verified the accuracy and consistency of terminology used in the documentation matched prior iterations.
- **Updating Documentation:** Revised existing documents to reflect changes in functionality and configuration of the ion implanter. Incorporate feedback from developer to enhance document quality.
- **Maintaining Documentation Standards:** Adhered to company style guides and documentation standards to ensure uniformity across all materials.

Related skills

Technical writing; Proofreading and editing; Documentation; Collaboration; Attention to detail; Information design; User-centric writing.

Job title: Peer Mentor
Employer: University of Massachusetts Amherst

(9/1/2019 – 12/1/2019) | Part-time

Summary of position and duties

My role as a peer mentor for the Computer & Information Sciences at UMASS Amherst consisted of regularly meeting with incoming students, informing them of opportunities, and checking in on their transition to college life. The goal of the mentorship is to help incoming students to better transition to the demanding requirements of the department and increase the retention rate of incoming students.

Responsibilities

- **Scheduling:** Coordinated meetings with 2 incoming students at least once a month to check-in on their wellbeing.
- **Communication:** Had casual conversations about activities within the Amherst area and within the department, as well as listened to any concerns they had.
- **Advising:** Highlighted important classes they should look into and their interests in the field to help direct them to opportunities that align with their interests.

Related skills

Interpersonal skills; Communication skills; Advising and guidance; scheduling; Networking; Leadership; Motivational Skills.

Awards and Honors

University of Massachusetts Amherst

2016	The Dean’s List, Fall Semester
2019	The Dean’s List, Fall Semester
2020	The Dean's List, Spring Semester

Notable Projects

DCTClock & Microtasks Drawing App

Description

At the request of Professor Ravi Karkar and Ph.D. candidate Abhay Anand, the project aimed to develop a drawing application for use with the iPad and Apple Pencil, capable of administering a series of microtasks (image completion and naming tasks) and the DCT Clock Test to participants. It collected metadata in JSON format from the Apple Pencil, saved the resulting images drawn by participants as PNG and JPG files, and stored timings on how long each task took. Additional functionality included the ability to upload collected metadata and images to Google Drive, the ability to input participant-specific information within a form page with text filtering at the start of the test, the ability to pause the test, and a dynamically changing UI.

Tech stack

- **SwiftUI:** Utilized for building the user interface of the drawing application, leveraging its declarative syntax to create a dynamic and responsive UI for the iPad.
- **PencilKit:** Library used to provide robust drawing capabilities, leveraging Apple Pencil for precise input. Such capabilities include the ability to capture high-resolution drawings and collect detailed stroke data, including pressure, tilt, and azimuth, which were then converted into JSON format for further analysis.
- **Xcode:** IDE responsible for enabling file sharing settings and building the application onto the iPad.
- **Python:** Used for data processing tasks, including regenerating of images from detailed stroke data and image comparison.

Earlyscreen Remote Data Pipeline

Description

At the request of SEED Labs, the project aimed to recreate their frustration game titled, *The Amazing Cake Game*, within the Unity Game Engine to provide support for Windows and iOS device (iPads and iPhones) support. The game's data collection features included collection of game event timings within SQLite tables, recording using the front-facing camera, and recording of the device microphone (Windows only). Data safety requirements implemented in the form of TLS 1.2 encryption and data deletion from device using Unity libraries.

The other component of the project involved implementation of the server-side application responsible for receiving the data from the application. Flask application is responsible for handling how to process GET, POST, and PUT requests with specific key values necessary for processing and saving encrypted data. Docker was used to containerize the Flask application on the Linux-based remote server, listening to specific ports.

Tech stack

- Unity Game Engine: Utilized 2D game development capabilities and frame-by-frame scripting to handle time stamps and game events of the recreated frustration game.
- C#: Used as the scripting language in the Unity Game Engine to implement game logic and manage game events.
- Xcode: Integrated Development Environment (IDE) used to deploy the built game onto TestFlight for remote distribution to study participants.
- SQL: SQL statements were run using C# libraries in the game scripts to store game events and timings in SQLite database files for use in separating frames and feedback windows.
- Python: Used for data processing tasks, including processing SQLite files containing timestamps and game events. Used in the implementation and analysis of the study's classification models.
- Docker: Containerized the Flask application hosted on the Linux-based remote server. Provides the Flask application with the necessary dependencies to function. Also utilized to run OpenFace software that was used for feature extraction on participant videos for use in classification models.

Penny-Pinching for Privacy: The Possibilities of Parsimonious Personalization

Description

Final Project for CS646: Information Retrieval where we sought to solve the problem of diversification in results produced by IR systems by performing redundant document removal on personal data corpus for personalized language model tasks. By removing redundant documents, we seek to prevent having the top *k* results in a search query characteristically similar. The underlying rationale behind our experiment is that there may be a level of bias due to uneven data distribution. The model that we used to test the results of redundant document removal was an Okapi BM-25 model. My primary role in the project involved implementing redundancy removal via abstraction of documents and placing them into groups via clustering algorithms. Total dataset was reduced by random removal within each group based on a percentage.

Tech stack

- Python: Utilized Python as the core language to program data processing and model building scripts.

- **Sci-kit Learn:** Used the library for clustering algorithms to separate the personal corpus of documents into groups based on “similarities”.

EleNa: Elevation Based Navigation

Description

Final Project for CS520: Theory and Practice of Software Engineering, where we aimed to implement a web application capable of finding the most optimal route based on elevation and distance. This involved implementing multiple path-searching algorithms, such as Dijkstra's algorithm, which would take in elevation and distance data across a series of points received from APIs to calculate the most optimal route and display it on the map. The user interface allows users to interact with the map element to select points of interest for navigation and choose their search criteria.

Tech stack

- **React:** Utilized for building the dynamic user interface, allowing users to interact with the map and select points of interest.
- **CSS/HTML:** Employed in structuring and styling the web pages to ensure a responsive and user-friendly design.
- **Node.js:** Served as the backend runtime environment, handling API requests to retrieve elevation and distance data from external sources.
- **Python:** Used to implement pathfinding algorithms and process the data received from Node.js backend to calculate the most optimal routes.

Music Genre Classifier

Description

Final project for CS685: Advanced Natural Language Processing, where we aimed to evaluate the performance of multiple models on the task of classifying music based on song lyrics. Tasks included data collection, data processing, model building, model tuning, and model performance evaluation. The rationale is that music can be classified based on their subject matter, leading to the task of semantic analysis of music lyrics. Models included in the comparative analysis include Top-K, BERT, Bidirectional Hierarchical Attention, and Convolutional Neural Networks.

Tech stack

- **Python:** Used as the primary programming language for all tasks, including data collection, processing, and model implementation due to its extensive libraries and ease of use for NLP tasks.
- **Pandas:** Utilized for data manipulation and analysis, allowing us to clean and preprocess the lyrics data efficiently.
- **Sci-kit Learn:** Leveraged for building and evaluating traditional machine learning models such as logistic regression, support vector machines, and decision trees. Provided tools for feature extraction, model evaluation, and hyperparameter tuning.
- **Google Colab:** Served as the development environment, enabling us to write and run Python code in a Jupyter notebook format. Provided free access to GPUs for faster model training and execution.
- **Tensorflow:** Used for building and training deep learning models, such as neural networks and convolutional neural networks (CNNs), to classify music lyrics. Provided high-level APIs for model development and deployment.
- **Pytorch:** Utilized for building and experimenting with more advanced and custom deep learning architectures due to its dynamic computation graph and ease of use for research and experimentation.

Intellectual Property Defense Simulation

Description

Final Project for CS564: Cyber Effects aimed to explore a documented exploit implemented within Metasploit by simulating its use case and implementing each step within an attack vector. This involved penetration testing, establishing persistence, exfiltrating data, and setting up a command-and-control (C2) server for communication. Specifically, we simulated an embedded PDF exploit that granted access to a terminal command line, enabling the execution of our payload.

The environment for the exploit involved social engineering tactics to persuade former employees to download and open the malicious PDF on their Windows XP devices. Upon opening the PDF, the payload was initiated, allowing communication back to our C2 server. This communication included checks for the existence of specific files, downloading said files, and removing any traces from the victim's device.

Tech stack

- **C#:** Utilized to create obfuscated data exfiltration scripts to locate the desired file from the simulated employee.
- **Metasploit:** Employed as the primary framework for developing and executing the exploit. Metasploit was used to create the payload that establishes a connection to the C2 server.
- **Bash Scripting:** Used for automating the malicious PDF generation and removal of exfiltration evidence upon task completion.
- **Flask:** Used to develop the C2 server. Flask handled incoming connections and would send obfuscated messages to the compromised system, communicating the payload on next steps.
- **Git:** Used to document the entire project for the class instructors. Allowed for the project team to coordinate progress on the project.
- **VMWare:** Used to create isolated virtual environments for testing the exploit. Windows XP virtual machines were set up to simulate the target environment safely. The attacker machine was a standard Ubuntu/Linux instance.
- **Apache2:** Configured to serve the malicious PDF file. Apache2 provided the necessary infrastructure to host the file and track download attempts.
- **FTP:** Used as a method for data exfiltration. The compromised system uploaded collected data to an FTP server set up to receive the information securely.