Jeremy Hsu

jeremyhsu.me | jeremyhsu@college.harvard.edu | github.com/hsujeremy

Education

Harvard University | B.A. in Computer Science — May '24

Selected Coursework Data Structures and Algorithms · Operating Systems · Database Systems · Computational Complexity Probability · Functional Programming · Design of Useful Interactive Systems · Linear Algebra and Differential Equations

Experience

Facebook | Software Engineer Intern – Facebook Avatars — Menlo Park, CA · Starting May '22

Lime | Software Engineer Intern – Partnerships Platform — San Francisco, CA · May '21 – Aug '21

- Developed a server-side API enabling seamless MaaS integrations between Lime vehicles and public transit providers which launched in August 2021 starting with riders in Berlin.
- Architected and implemented the end-to-end refund flow along with the lifecycle APIs for users and for fetching real-time vehicle data through QR code scanning or license plate entry.
- Extended the functionality of a company-wide internal administrative portal by creating UI components that enable assigning permission access-control roles and viewing mobility data feed instructions for regions and external partners.

Cisco | Software Engineer Intern – Webex Media Engine — San Jose, CA · May '20 – May '21

- Overhauled and extended the team's internal debugging and triage tools by writing automation, analysis, and visualization scripts, and deploying them with custom-built Jenkins jobs and CLI commands, saving engineers multiple hours per day.
- Collaborated with overseas teams to develop and deploy a data aggregation module leveraging the Elastic API that resulted in Kibana dashboards loading up to 5x faster.
- Created and monitored interactive dashboards visualizing critical media quality and network performance metrics, enabling engineers to identify problematic trends and issues more rapidly.

Activities

Harvard Data Systems Lab | Research Assistant — Cambridge, MA · Feb '22 – Present

- Advised by Prof. Stratos Idreos and Utku Sirin on designing CascadeNets, a cascading classifier composed of lightweight CNNs which aims to reduce training time, improve model adaptability, and address subproblem heterogeneity.
- Implemented an optimal subtask partitioning algorithm based on centroid distances with k-means clustering and benchmarked a custom ResNet-based model to determine the optimal hyperparameters for different partitions of the ImageNet dataset.

Datamatch | Algorithm Lead — Cambridge, MA · Sep '19 – May '22

- Redesigned and rewrote significant portions of Datamatch's scoring and stable matching algorithms and added flexible match balancing, all of which improved overall performance, usability, and extensibility.
- Improved Datamatch's pairwise compatibility scoring function by implementing sentence embeddings with Sentence-BERT to analyze user sentiment similarities, and dynamic inverse-proportional weightings to address answer distribution polarities.

Projects

CoffeeOS — C++, ASM

CoffeeOS is a multicore x86-64 operating system that supports kernel task suspension, system calls, virtual memory, and multithreading. Additional features include wait queues, caching, prefetching, synchronization primitives such as spinlocks and futexes, and an on-disk file system supporting file extensions and directory trees.

ShuttleDB — C

ShuttleDB is a NoSQL column-store database supporting fast select-project-join queries on columns with up to 10M tuples. In addition to the parser, optimizer, and executor, ShuttleDB is persistent on disk and includes internal query optimizations such as B+ tree indexing, custom TCP streaming protocols, prefetching, zone maps, and multithreaded scan sharing.

YouTube Party — Node.js · React · Socket.IO · YouTube iFrame API

YouTube Party is a full-stack web app that allows any number of users to stream YouTube videos synchronously using WebSockets. YouTube Party synchronizes pause and play operations and includes an in-app chatroom.

ML-Enabled Spotify Curator — Flask · React · Celery · Redis · Scikit-Learn · Spotify Web API

Spotify Curator is a full-stack web app that predicts whether a user will like a particular song based on musical characteristics of songs in their Liked Songs library. Spotify Curator uses a random forest classifier on the backend supported by a distributed task queue to enable non-blocking client requests.

Technical Skills

Programming Languages C · C++ · Java · Python · JavaScript · Swift

Frameworks & Technologies React · Ruby on Rails · Node.js · TensorFlow · Keras · SwiftUI · Firebase · Elasticsearch