SUJIT PACKIARAJ

425-647-0715
sujit.pack@gmail.com
linkedin.com/in/sujitpackiaraj
github.com/sujpac

SKILLS

Programming Languages
Python, Java (preferred)
C++, Go, JavaScript, HTML5, CSS3

Tools & Frameworks
Django, Pygame, Docker, Spring, PyTorch, Numpy, Pandas, Scikit-learn, React.js, Node.js

Distributed Systems & Big Data
Postgres, MySQL, MongoDB, Amazon DynamoDB, SQL vs. NoSQL, MapReduce, Hadoop, Redis, Memcached, Amazon SageMaker, Digital Ocean, Heroku, AWS

Software Engineering
Backend Development, Testing, Automation, Deployment, API Development (REST & GraphQL), Python Development, Java Development, Object-Oriented Design, System Design

PROJECTS

Film Review Sentiment Analysis
April 2020 - June 2020
- Constructed a recurrent neural network to guess the sentiment of film reviews
- Trained model on the IMDb dataset using PyTorch hosted on Amazon SageMaker
- Wrote a lightweight web app to interact with the deployed SageMaker model

EDUCATION

B.S. Computer Science
University of Washington
Seattle, WA

Coursework
Operating Systems, Distributed Systems, Database Systems, Compilers, Algorithms, Artificial Intelligence, Machine Learning

WORK EXPERIENCE

Amazon — Software Engineer 2017

- Developed a new backend infrastructure for the Seller Support app — help desk software used by over 300,000 Amazon sellers worldwide as well as by various teams internally (including AWS) to establish a direct line of communication between customers and Amazon representatives to resolve queries and issues
- Utilized Amazon DynamoDB for storage of over **10TB** of data (previously stored via Oracle) while increasing data processing speeds by over **30%**
- Directly saved over **$50k** just by allowing Amazon to drop Oracle
- Modified 100+ live API backend endpoints (receiving over 10M requests per day) to interface with the new DynamoDB backend with zero disruption of service or downtime
- Technology Used: **Java**, **Spring**, **DynamoDB**

Facebook — Software Engineering Intern 2014

- Developed a new feature for Facebook Ads called Page Engagement Targeting — allowing page owners to target 'recently engaged' users (liked, commented, viewed, or interacted with page content within the past 90 days) for ads
- New feature was **2x as effective** (measured via clickthrough rate) as the sole existing feature for page owners at the time which targets the page's fans & the friends of those fans
- Developed logging infrastructure in Hack to log 'recent engagement' with pages
- Wrote C++ backend code to automate processes which aggregated (terabytes) of raw log-form data, interpolating them into a simple data model that showed for a given page whether a given user has 'recently engaged'
- Added JavaScript UI code to make the feature visible to advertisers
- Technology Used: **Hack** (PHP dialect), **JavaScript**, **React.js**, **C++**, Folly

University of Washington CSE — Teaching Assistant 2014

- Planned my students' progress through a rigorous 3-month long curriculum of an intermediate Programming Languages course. Guided the growth of their relatively unsophisticated, Java-centric understanding of programming to a richer, multi-language understanding of programming paradigms (e.g., higher-order functions, lexical vs. dynamic scoping, evaluation, etc.). In the process, my students gained fluency in 4 new and diverse programming languages — Racket (Lisp dialect), **Haskell**, Prolog, and **Ruby** — a foundation upon which any other programming language could be added with ease.

Expedia — Software Engineering Intern 2013

- Built automated deployment verification tool (which scanned logs to verify that the latest deployment didn't introduce errors) for the lodging partner services inventory app that processes real-time inventory updates
- Tool reduced the time engineers spent on deployment verification by **75%**
- Technology Used: **Java**, Scala, **JavaScript**, Play Framework, **Splunk** API

University of Washington Networks Lab — Research Assistant 2012

- Developed and ran a censorship detection experiment — querying several thousand potentially sensitive domains taken from known blacklists (such WikiLeaks) against publicly available DNS servers in 59 countries — to quantify the extent to which countries implement DNS blocking.
- **Co-authored publication** in Tiny Transactions on Computer Science, Vol.2
Detecting DNS Censorship without an internal vantage point
- Technology Used: **Python**, Tornado, **MySQL**, Bash, PlanetLab