Zhi Ji

1 Morningside Drive New York, NY 10025 | 646-897-8852 | zj2242@columbia.edu
https://github.com/zhiji95 | https://www.linkedin.com/in/zhiji1019/ | https://zhiji95.github.io/

EDUCATION BACKGROUND

Columbia University *New York, NY* *(Expected) 09/2018 - 12/2019*
MS in Electrical Engineering, GPA: 3.6/4.0
Relevant Courses: Algorithms, Operating System, Database, Cloud Computing, Deep Learning, NLP, Distributed Network

University of California Berkeley *Berkeley, CA* *01/2017 - 12/2017*
Exchange Student in Electrical Engineering & Computer Science, GPA: 3.6/4.0
Relevant Courses: Data Structure, Machine Learning, Communication Network

University of Electronic Science & Technology of China *Chengdu, CN* *09/2014 - 07/2018*
BS in Electronic Information Engineering, GPA: 3.8/4.0
Relevant Courses: C Language, Software Fundamentals, Computer System, Data Analysis

PROFESSIONAL SKILLS

- Technical Skills: Python, Java, JavaScript, C, C++, SQL, HTML, CSS, Scala, TensorFlow, Keras, Spark, Kafka, Linux kernel.
- Web Development: React, jQuery, Sass, DOM, AJAX, Node.js, Flask, Django, MongoDB, AWS, Docker, Akka, Play.

PROFESSIONAL EXPERIENCE

Walmart Labs/ Jet | *Software Development Engineer* *Hoboken, NJ* *06/2019 – 08/2019*
- Worked as a full stack web developer and added 5 features for Parcel delivery web application frontend using React.js, Redux, JSX, jQuery, Sass, Django and implemented backend RESTful APIs using Flask, SQL, Docker and Kafka.
- Managed MySQL database and modified tables using Marshmallow and SQLAlchemy.
- Wrote tests, built and deployed updated applications using Docker, Azure and Jenkins.
- Built the barcode scanning application for all drivers to use and integrated with the rest of Parcel application workflow.
- Worked on route optimization system for fulfillment center using Akka, Play framework and Kafka in Scala.

Chinese Academy of Sciences | *Machine Learning Engineer* *Beijing, CN* *04/2018 - 08/2018*
- Developed several machine learning and deep learning models for stock price prediction and trading strategies using Tensorflow.
- Built the backend in Node.js and Flask, front end in Django, jQuery and D3.js for data visualization.
- Utilized Kafka, Spark and MLlib to process raw stock prices data.
- Researched on the algorithm for detecting black product attack with imbalanced sample distribution and missing features.
- Published a paper at CSAE 2018: https://dl.acm.org/citation.cfm?id=3277966

Berkeley Video and Image Processing Lab | *Research Assistant* *Berkeley, CA* *05/2017 - 12/2017*
- Developed a sensor-based sorghum height and width estimation algorithm with Fast RCNN.
- Programmed in C++ to control and adjust Intel RealSense camera parameters for data collection.
- Published at Electronic Imaging 2018: http://www-video.eecs.berkeley.edu/papers/jihui-jin/jihui-height-ei-2018.pdf.

PROJECT EXPERIENCE

Let's Meet *Columbia University* *02/2019 – 05/2019*
- Built a web service for group meetup that recommends movies, restaurants and shows attendee's location and a chatting room.
- Setup APIs using API Gateway with lambda functions and authentication using Cognito.
- Applied React.js to implement user interface and hosted the frontend in an AWS S3 bucket.
- Built the SQS, SES, DynamoDB and Elastic Search with machine learning for restaurant recommendation and email notification.
- Built an Android application communication with AWS services.

Operating System *Columbia University* *01/2019 - 05/2019*
- Built different types of HTTP multi-thread server in C, including multi-processes, multi-threads and Nonblocking I/O.
- Developed and added the customized multi-core scheduler with CPU thread group affinity to Linux Kernel.
- Built an in-memory Linux File System from scratch.

Computer Vision *Columbia University* *10/2018 - 12/2018*
- Designed and trained the conditional Deep Convolutional Generative Adversarial Networks with classifier for human face images completion and classification with high performance.
- Refined a deep Q-learning algorithm for image restoration by Double Q-learning, Prioritized Replay and Dueling Q-learning.