

Tejas Ruikar

truikar@asu.edu | +1 480 381 0007 | [in linkedin.com/tejasruikar](https://www.linkedin.com/in/tejasruikar)

Education

Arizona State University, Tempe, Arizona (GPA- 3.89/4.0) Intended Graduate May 2020
Courses: Artificial Intelligence, Fundamentals of Statistical Learning, Foundations of Algorithms, Statistical Machine Learning, Data Mining, Data Visualisation, Distributed and Parallel Data Systems
College of Engineering Pune, Pune, India (COEP) August 2012 - May 2016
Bachelor's in Computer Science and Information Technology (GPA - 3.6/4.0)
Relevant Courses: Cloud and Big Data, Storage and Virtualization, System Programming and Operating Systems.

Skills:

- | | | |
|---|--|--|
| • Languages: JAVA, Python, C#, C, C++ | | Web Technologies: ReactJs, Spring, JavaScript, MEAN Stack, |
| • Databases: MySQL, MongoDB | | Simulation Software: Qualnet |
| • Tools: Visual Studio, IntelliJ, Eclipse | | Other: MSOffice, TeamCity, Odyssey |
| • | | |

Employment

SDE Intern **Amazon** May 2019- August 2019
FOSUI

- Successfully completed a full-stack project from scratch which serves the purpose of being the face of the next generation Fulfillment Operating System at Amazon.
- The project was completed using ReactJs for frontend and Java Spring server at backend to fetch and post the data towards FOS in order to manage fulfillment resources. Successfully demonstrated the project which would be taken over by a dedicated team soon.

Technology Analyst **Credit Suisse** August 2016- July 2018

Successfully completed Technology Analyst program.

MiFID II ShortCode-LongCode delivery

- Implemented a Java-Spring architecture that fetches data from multiple data sources with over 10^7 entries, normalizes them and makes data in exchange specific formats (CSV, XML etc) and delivers them via SFTP, REST or APIs given by exchanges, gets the response and persists it in the database.
- MiFID II is mandatory regulation in Europe and all the trading entities were bound to be compliant by 3rd January.

Global Order Management System (GOMAN)

- GOMAN is a software that is extensively used in Credit-Suisse for high volume trading across the globe. Worked in the development team for GOMAN where there were daily requirements and bi-weekly deployments. GOMAN architecture is in .Net C# and MySQL database.

Intern **Tata Consultancy Service** May 2015-July 2015

Project Management Dashboard: The application helps managers to Add, Update, Delete the project data, maintain employee profiles and a sign various projects and tasks to them. The application was developed in MEAN stack.

Publication

Cluster Based Hierarchical Addressing for Dynamic Source Routing in Ad-Hoc Networks Springers 2016

- The paper proposes an algorithm that divides a WANET into a tree and assigns hierarchical addresses which results into flood reduction and finding one direct route to the destination node increasing the efficiency and reducing the power consumption of every node.
- The algorithm was simulated in Qualnet by modifying code existing code for DSR in C++ to add addressing algorithm to it and demonstrated reduction in flooding by 79%.

Academic Projects:

Pac-Man AI – A running implementation of AI Techniques: - Fall 2018

- Implemented Pac-Man agents using Machine Learning, A* search, particle filtering, Alpha Beta Minimax and Expectimax and Reinforcement learning in Python.

Implemented D* Lite Algorithm in Pacman Project- LifeLong Planning Fall 2018

- Successfully implemented LPA* and D*Lite algorithms to find shortest path in PacMan AI project. The implementation was done in Python.

Modelling Annotated Data :- Fall 2018

- Implemented Corr-LDA from the paper Modelling Annotated Data by Andrew NG. The Core15k data was trained for image captions using its SIFT features. The caption perplexity was measured for testing data against training data. The algorithm was tested against various databases to verify the results were consistent and reliable.

Implemented 1-Hidden layer Neural Network, Fully connected 3-Hidden Layer Neural Network, Kernal Soft-Margin SVM, Kmeans algorithm, GMM with EM algorithm. All the algorithms were implemented using numpy and scikitlearn.