






ROHITH SATHIAMOORTHY PANDIAN

 [rohithaug.github.io](https://github.com/rohithaug)  linkedin.com/in/rohithsp  github.com/rohithaug  858-319-6968  rsp223@ucsd.edu

Education

University of California San Diego

California, US

Master of Science in Computer Science (CGPA: 4.0/4.0)

Sep 2022 - Jun 2024

Coursework: Networked Systems, Algorithm Design and Analysis, AI - Probabilistic Reasoning and Decision Making, Recommender Systems and Web Mining, Search and Optimization, Biomedical Natural Language Processing

PSG College of Technology, Anna University

Coimbatore, India

Bachelor of Engineering in Robotics and Automation

Jul 2016 - Sep 2020

CGPA: 9.91/10.0, University Rank: 1 (Best Outgoing Student, Gold Medallist)

Technical Skills

- **Languages:** Python, JavaScript, TypeScript, Golang, JAVA, C, C++, HTML and CSS, SQL, PowerShell
- **Frameworks:** Node.js, React, Redux, REST, Tensorflow, Scikit, MongoDB, D3.js, Keras, Pandas, NLTK
- **Tools:** JIRA, GIT, Tableau, Jupyter, Elastic, Kibana, Logstash, Postman, Jest, Docker, Minitab
- **Platforms:** Windows, Linux, IIS Web Server, Microsoft SQL Server, Arduino, Raspberry

Professional Experience

Ramco Systems | Analyst Programmer - New Initiatives | Chennai, India

Nov 2020 - Jun 2022

- Tools: **React, Node, JIRA, SQL, REST APIs, MongoDB, IIS Server, ELK, D3.js, OIDC, PowerShell**
- Full Stack Development, Data Science, API Integration, Distributed Systems, Deployment Automation
- Led development of a semi-automated customer support system for creating tickets from the Ramco Core Software, improving fix generation for support teams. The system reduced overall the Service Level Agreement by 30%, improved customer self-service and is currently live for over 60 customers.
- Integrated system logs to elastic and designed Kibana dashboards to monitor performance & user metrics.
- Implemented a classification model for Ticket Type and Sub-type prediction using Supervised Machine Learning and later improvised using BERT Transformer Model and achieved an accuracy of 86%.
- Developed a module to dynamically suggest help content based on the Product Business Component, Screen Context, and User Issue Summary from the company documentation site using Solr Search.
- Automated the deployment pipeline for the system to scale feature release for 40+ public cloud customers.
- Built 25+ and reviewed 40+ React UI components for the Ramco Low Code Platform.
- Designed a scheduler to sync time bookings of the employees from JIRA and Wrike to the internal timesheet application to facilitate tracking the effort spent on various projects and support activities.

Projects

Pathfinding and Maze Algorithms Visualizer | Web Development and Graph Traversal

[Project](#) | [Code](#)

- Built a react based static web page to visualize Path finding (Dijkstra's, A*, Bidirectional Greedy, Breadth-First) and Maze Generation Algorithms (Recursive Division, Vertical Division, Random Maze).

TritonHTTP Web Server | Golang and HTTP/1.1

- Implemented a concurrent, mini web server that conforms to a subset of the HTTP/1.1 protocol.

Representation Learning for Sepsis Prediction | Deep Learning and Natural Language Processing

[Code](#)

- Utilized Deep Learning Models (CNN and LSTM) to predict sepsis in patients admitted to the ICU based on MIMIC-IV chest X-ray reports and clinical reports. Improved performance by incorporating a 200D PubMed Word2Vec embedding, resulting in a positive predictive value of 80%.

Exploring the Interplay of Ratings and Categories in Google Local Reviews | Recommendation

[Code](#)

- Investigated the relationship between restaurant information, GPS, and user review sentiments in predicting ratings and personalized cuisine recommendations using the Google local reviews dataset. The proposed strategy exhibits an accuracy of 89% or less in prediction experiments.

Sorting Algorithms Visualizer | Python, PyGame

[Project](#) | [Code](#)

- Developed a graphical user interface to visualize sorting algorithms (Quick, Bubble, Heap, Radix Sort, etc).