321 N Fremont Ave, #115
Tucson AZ 85719

SRIDHAR MOCHERLA

srmocher.github.io

(520) 225-7243
srmocher@email.arizona.edu

EMPLOYMENT

Research Assistant **Steward Observatory** **Aug 2016 - current**

Wikipedia Classification

- Implemented efficient XML parser for Wikipedia dump (50GB+) with ~3 hours runtime.
- Implemented Naïve Bayes and SVM classifiers using Java/Python to classify Astronomy articles with 80% and 85% accuracy respectively
- Implemented on-the-fly indexing of classified articles into Solr service exposed to front-end

Software Engineer **Microsoft** **August 2015 – May 2016**

Enterprise Integration Platform

- Developed PaaS solutions on Azure as part of agile scrum team to built B2B platform
- Enhanced performance of Tracking REST API (fixed timeout) by leveraging connected databases on cloud.
- Modularized deployment automation framework of platform components for faster release to production

Service Engineer **Microsoft** **July 2014 – July 2015**

OEM platform

- Reduced downtime by automation (by 30%) of different phases of on-premise disaster recovery using PowerShell scripting
- Built mobile app (Windows Phone) for monitoring of services deployed on cloud.

EDUCATION

Tucson, AZ **University of Arizona** **Fall 2016– May 2018**

- Master's in Computer Science (M.S). GPA: 3.67/4. Full tuition scholarship.
- Notable coursework: Computer Vision, Adv. Data Visualization, Database Systems Implementation

Hyderabad, India **Osmania University** **Oct 2010 – June 2014**

- Bachelor of Engineering (B.E) in Computer Science. Aggregate: 88.4%
- Awarded silver medal for the academic year 2012-13 for being the top student in the department

TECHNICAL EXPERIENCE

Projects

- **Minibase** (Spring 2017)– Built features for a mini-relational database in C++ – Heap file with CRUD operations, Buffer Manager with replacement policies, B+-tree index and sort-merge join of files.
- **Simple Router** (Fall 2016) – Virtual router built in C with support for file download, ping, traceroute, IP packet forwarding, ARP caching and multi-router support with link-state routing.
- **Fluid Simulation** (Spring 2017) – Physics based fluid simulation rendering real-time using OpenGL and CUDA. Water-like fluid generated as particles and surface display with motion.
- **Medical data visualization** (Spring 2017) – Tooth data visualized as a surface using Marching Cubes algorithm in C++ using openFrameworks.

Languages and Technologies

- C++, C, Java, C#, SQL, JavaScript, Angular, PowerShell, MATLAB, OpenGL
- Visual Studio, Microsoft SQL Server, CLion, git