Avinash Shanker

Austin, Texas **Contact**: +1-682-401-9365

Email: savinash273@gmail.com **LinkedIn**: https://www.linkedin.com/in/avishanker/ **GitHub**: https://github.com/avinash273

Skills

Languages: Java, Python, JavaScript, SQL, PL/SQL, Shell Scripting, C++
Web Technologies: HTML, CSS, PHP, Android, iOS, Spring, Angular, NodeJS, ReactJS, PostgreSQL, Apache Spark, Hadoop
Libraries and Tools: Keras, TensorFlow, Google Colab, OpenCV, Open MPI, JUnit, AWS, GitHub, Jenkins, Docker, Splunk

Education

Master of Science in Computer Science **Aug 2018 – May 2020**
University of Texas Arlington
Dual Specialization: Artificial Intelligence *(Neural Network, Computer Vision)*, Database Systems *(Hadoop, Spark, AWS)*

Bachelor of Technology in Information Technology **Aug 2011 – Mar 2015**
National Institute of Technology, Karnataka

Experience

Software Engineer, TESCO *(Java, JavaScript, Spring, MongoDB, PL-SQL, ProC, AngularJS, Jenkins)* **Jul 2015 – Jul 2018**

- Developed RESTful web service using Spring framework and MongoDB to monitor and re-process orders which failed to integrate in Store system in agile team. Tested API using Postman and JUnit. Used AngularJS UI to consume the service and provide feedback on order rejection status. This enabled to easily correct orders, reducing manual effort on integrating to store from 6hrs/week to 0.5hrs/week. Used GitHub for version control, JIRA for project management
- Designed a multithreaded mechanism to enhance the performance of existing auto-ranging functionality in Product Information Management (PIM) system. Auto-ranging assists in ranging 50 million products to newly setup warehouse. To expedite, distributed load evenly to 10-threads by hashing product ID. Worked on SQL Sever & PL/SQL
- Developed REST API using Spring framework to validate, daily sales with current stock on hand. Any discrepancy in expected stock, enabled Stock Check API to flag such products in the DB for manual review and generating notifications on dashboard. This enhanced the response time for stock discrepancy correction by 2hrs

Projects

Big Data Hadoop on San Diego Supercomputer Center *(Java, Scala)* **Mar 2020 – Apr 2020**
- Analyzed connected components of extremely large directed graph using Hadoop's HDFS and Map-Reduce framework as maven implementation. For each node, computed number of node neighbors and grouped nodes by neighbors.
- Implemented Lloyd's K-Means clustering algorithm, to partition coordinates into K clusters of neighboring coordinates

Deep Privacy Face De-Identification Using GAN *(Python, TensorFlow, Keras, OpenCV)* **Nov 2019 – Jan 2020**
- Ensure anonymity for faces in public images, used Generative Adversarial Network (GAN) and Autoencoders to generate a highly realistic anonymous face mask. Used Kalman filter for smoothing boundaries of frame and MTCNN face detection algorithm to reconstruct superimposed face on Google Colab decreasing reconstruction loss to 30.02%

Analysis on Drug Consumption Dataset *(Python, Scikit, Pandas, NumPy, Seaborn)* **Jan 2019 – Mar 2019**
- Performed K-Means and Hierarchical Agglomerative clustering algorithm from scratch to train, predict & contrast performance of both models using confusion matrix (precision & recall) and classify drugs used by different age groups
- Visualized the data with 92.3% accuracy in K-means and 79% in agglomerative clustering. Improved the K-means accuracy by 4.2% isolating elements that are very far above the cluster's variance threshold value

Parallel Sieve of Eratosthenes for Prime Generation *(Open MPI, C)* **Dec 2018 – Jan 2019**
- Implemented Sieve algorithm for prime number generation as a parallel program on Stampede Supercomputer.
- Generated primes under 10^9 using 32 parallel processors in 0.074908 sec, improved the performance of algorithm by making greedy choice in assigning the initial prime number for each processor to mark multiples and counting array indexes to obtain primes. The original algorithm took 0.085918 sec for 10^9 primes making it 3.7% faster

Honors and Awards

- Awarded biannual Tesco Star Performer twice in a span of 3 years among a team of 22 members for creating a UI for order reject re-processing and performance tuning of auto-ranging batch within 4 days for going live in Poland stores.