KRAPI SHAH

krashah@cs.stonybrook.edu | +1 631 816 5086 | Stony Brook, New York - 11790
LinkedIn: www.linkedin.com/in/krapi-shah | GitHub: https://github.com/shahkrapi

EDUCATION

State University of New York, Stony Brook, USA (GPA: 3.92/4.00) **August 2018 - Present**
Masters in Computer and Information Sciences Expected Graduation: December 2019
Current Courses: Machine Learning, Data Science Fundamentals, Operating Systems, Cryptography, Analysis of Algorithms

Veermata Jijabai Technological Institute (VJTI), Mumbai, India (GPA: 9.17/10.00) **July 2013 - May 2017**
Bachelors of Technology in Information Technology with *Distinction*. *Ranked 5th in class of 71.*
Relevant Coursework: Data structures and Algorithms, Artificial Intelligence, Database Fundamentals, Web Technologies, Distributed Systems, Software Engineering, Operating Systems, Data Mining, Network security

SKILLS

- **Programming Skills :** Python, Java, C++, AWS, Google Cloud
- **Web Development :** Bootstrap, HTML, CSS, Javascript, Jinja, Flask, Gunicorn
- **Frameworks/Libraries** : LibSVM, PyTorch, TensorFlow, Sklearn, Java Design Pattern
- **Tools :** Git, JIRA, BitBucket

PROFESSIONAL EXPERIENCE

Technology Intern *at Tradeweb, Jersey City, USA* **June 2019 - Present**

- Designed and developed a web based tool for application monitoring across Development, QA, and Production environments using Python, Flask and Jinja.
- Enhanced the existing command line tool supporting development environments to make it accessible over the web for non development environments and achieve 1-click central access to all application details.
- Collaborated with a team of 4 to benchmark performance and security characteristics for AWS data streaming services like Amazon Kinesis, SNS, SQS to facilitate streaming financial data to clients over the cloud.

Technology Analyst *at Citi, Pune, India* **July 2017 - July 2018**

- **I**mplemented an Automated File Generation Framework to generate thousands of files of various formats CSV, flat-files, XML within seconds for load testing of the payment processor application.
- Performed root cause analysis to identify problems that led to the failure of the update of the security matrix for maker-checker functionalities leading to a delay of 3 weeks for each failure. Developed a program that automates the update and verifies the security matrix before generating the script for the change**.**
- Incorporated Mockito to automate JUnit testing of payment processor application and achieve 85% JUnit coverage.

Technology Intern *at Citi, Pune, India* **May 2016 - July 2016**

- Introduced the use of Apache Camel as a middle layer to ease interaction within various interaction points of payments systems like client portal, ERP file, message queues.
- Increased code reusability and maintenance for message transformation tool using Apache Camel and web services by replacing 5 different services for different endpoints by 1 common service for all endpoints.

PROJECTS AND PAPERS

Multi-Tier Caching Analysis for Cost and Performance Optimization **January 2019 - Present**
File Systems and Storage Lab, Prof. Erez Zadok

- Analyzing storage workload traces using caching simulators to optimize overall storage costs against parameters like throughput, energy consumption, and hardware costs for different physical devices at different layers

Action Recognition Using RNN **October 2018 - November 2018**

- Implemented an architecture consisting of linear layer followed by two LSTM layers with ReLU activation functions and Cross Entropy Loss to achieve an accuracy of 81.37%.

Google Analytics Customer Review **September 2018 -October 2018**

- Analysed the Google Analytics Data for Google Store to predict the revenue generated from each user using Random Forest Regressor and XGBoost using Python.

Solving Rubik's Cube Using Graph Theory **June 2016 - May 2017**

- Developed a Rubik's Cube solver using bidirectional search algorithm in C++ by adapted efficient memory management techniques along with Rubik's cube properties of cube symmetries and antisymmetries to efficiently traverse the possible search space of 43 quintillion.
- Paper Link : https://link.springer.com/chapter/10.1007/978-981-13-1132-1_24