Takshshila Rawat

480-335-1908 Mail: trawat2@asu.edu LinkedIn: takshshila Github: takshshila Tempe, Arizona

Education

- **Arizona State University, Arizona** **Aug 2021 – May 2023**
  Master of Science in Computer Science ***CGPA: 3.96/4***
- **National Institute of Technology Hamirpur, India** **Aug 2012 – May 2016**
  Bachelor of Technology in Computer Science ***CGPA: 7.56/10***

Skills Summary

- **Programming** Python, Cuda-Programming, Java, AngularJS, JavaScript, HTML, CSS
- **ML Technologies** PyTorch, Scikit-Learn,TensorFlow, Keras, NumPy, Pandas, NLTK, Networkx
- **Tools and libs** Jupyter Labs, AWS (EC2, Lambda, S3, SQS, DynamoDB), D3, HuggingFace
- **DevOps Technologies** Git, Jenkins, MySQL, Docker, Flask, Postman, Neo4j, Elasticsearch

Professional Experience

- **Machine Learning Research Aide - [Python, Cuda, MQTT, Pandas, CNN, Flask] W P Carey** June-22 - Current
  - Developed a modular system using a Self-Organing map and IOT as an alternative to decision trees, gradient boosting, etc.
  - Optimized computational resources using effective utilization of GPU by Cuda programming
  - Trained and analyzed self-organizing maps on static and streaming data to make the system robust.
  - Performed real-time edge computing on Nvidia Jetson and Ngx to simulate edge devices.
  - Worked on explainable AI for real-time object monitoring on the XAI dataset
- **Senior Software Developer - Reliance Jio, India** April-19 - August-21
  - **Integrated Performance Manager - [Java, Kafka, AngularJS, HTML, Elasticsearch]**
    - Led a team of 10 people to develop a 5G service that predicts and monitors real-time network performance
    - Project consisting of six microservices replaced other vendor services.
  - **Machine Learning as a Service - [Python, AngularJS, HTML, Elasticsearch]**
    - Managed a team of 8 to implement a service consisting of an Anomaly Detection and Forecasting Engine on real-time data
    - Used SVM, Random Forest, Gradient Boosting, etc. in an ensemble way that imporve performance and reduced work by 70%
- **Software Developer - Reliance Jio, India** June-16 - March-19
  - **Capacity Manager - [Java, Kafka, AngularJS, HTML, Elasticsearch]**
    - Developed a microservice to monitor and autoscale resources(CPU, RAM, Docker containers, VNF) for NFV/SDN (Network Function Virtualization and Software defined network) cloud
    - Improved performance of the NFV/SDN cloud by 45%
  - **Adaptive Troubleshooting and Operation Manager - [Java, AngularJS, Elasticsearch, Neo4j]**
    - Introduced a platform for analyzing, monitoring, and troubleshooting real-time call/message data records.
    - Data collecting, preprocessing, analysing, and displaying 92 million users' data as How India Talks
  - **Centralised Command Line Interface - [Java, apache-commons, jline, POI, Elasticsearch, jetty]**
    - Designed and developed a centralized and dynamic CLI (alternative to the user interface) for monitoring FCAPS.
    - The single-person project reduced ≈ 90% of the workload and is used by ≈ 95% of the company projects

Project

- **Dialog system: Hierarchal Help Me Think -** [PyTorch, HuggingFace, Pandas]
  - Explored multiple prompting techniques on GPT-3, Flan-T5, XLNET, GPT2, etc. to help non-expert users to solve any task
  - Developed custom dataset and fine-tuned the models to generate hierarchal help-me-think tasks comparable to GPT-3
- **Understanding Indirect question and answers -** [PyTorch, HuggingFace, Pandas]
  - Fine-tuned BERT model for Natural Language Understanding on MNLI, BOOLQ, and Circa dataset
  - Replicated experimental table for relaxed labels and reached comparable accuracies.
- **Compressing BERT -** [PyTorch, HuggingFace, Pandas]
  - Fine-tuned and compressed bert-base model using Pruning, Quantization, and Knowledge distillation
  - Decreased time by ≈12x and size by ≈10x on MNLI dataset.
- **Face Recognition on Raspberry Pi using AWS -** [PyTorch, Python, EC2, S3, DynmoDB, Lambda]
  - Developed an IAAS and PAAS service for real-time face recognition system
  - Perform recognition with 98% training accuracy and 100% validation accuracy
- **Understanding Text Classifier using Counterfactual Explanation -** [Python, Tensorflow, Pandas, BERT]
  - Compared causal models: Deep, Cxplain, and AAAI models on multiple datasets
  - Interpret models with LIME, SHAP, and Unified Information Explainer
- **Pointer Generator Text Summarization -** [Python, PyTorch, Pandas, Transformers]
  - Implemented seq-to-seq encoder-decoder with attention architecture for pointer generator text summarization
  - Used CNN/Dailymail datasets for abstractive and extractive text summarization