Anshita Saxena

anshita333saxena@gmail.com +1 (514) 569-5243 Anshita Saxena

Work Experience

Ericsson Global AI Accelerator Lab Montreal, Canada
Machine Learning Engineer (Jan 2024 - Present)

- Conducting a study focused on distributed machine learning and federated learning.
- Designing methodology to build a communication-efficient network topology that reduces the trade-off between convergence and latency per iteration.
- Developed end-to-end RAG based LLM application which got accepted as part of Ericsson Developer Conference 2024.

Hydroquebec Research Institute (IREQ) Montreal, Canada
Applied Deep Learning Researcher (May 2023 - Dec 2023)

- Conducted a Research Project focused on developing a virtual sensor for estimating displacements in high-frequency turbine labyrinth seals. This involved leveraging Deep Learning techniques to extrapolate high-frequency characteristics from low-frequency measurements.
- Proposed and developed a novel and successful way to employ the WaveNet model, which is generally used for Speech Generation, for generating signals from a rotating machinery (time-series dataset) to address a complex problem statement characterized by diverse information sources operating at various levels. Libraries such as ydata-synthetic to apply TimeGAN and run on a super-computer (cluster of GPU nodes). Improvements done on BiLSTM.
- Used MAE as a loss function, initial training loss was 7.5 which was reduced to 0.7. Generated signals overlapped the target signals for about 89% of test data. All models were implemented in PyTorch, JAX, and Keras. Total of 4 million datapoints for high-frequency data in comparison to 10k datapoints for low-frequency data.

IBM Bengaluru, India
Data Scientist/Software Engineer - Big Data (Apr 2018 - Aug 2022)

- Orchestrated the end-to-end pipeline for a loan prediction machine learning model on our AI Data Platform, showcasing a seamless process from ingesting raw data into S3 buckets to model construction. Leveraging PySpark and Azure Machine Learning Pipeline, the models, including Random Forest and XG-Boost, were meticulously built. Visualization using Dash Plotly provided comprehensive insights. The implementation covered data from 8 different bank clients with 1.2k customers, where we built the Canonical Model to refine data into one format, aiding in the identification of potential loan defaulter customers using Random Forest and XG-Boost models. Additionally, a White Paper for the Practice Department was authored, responding to a request from senior leadership.
- Developed microservice APIs for Kafka ingestion and NoSQL database retention. Processed 2600 messages/min at 500 concurrency level in a multi-datacenter mode.
- Innovatively developed a microservice API on Kubernetes, facilitating the transmission of error and exception messages from model microservices to a Slack channel. This not only streamlined early alerts and remediation but also significantly reduced the monitoring workload by 60 hours.
- Contributed significantly to cost savings by developing automation scripts for data extraction, manipulation, and reposting into Kafka using Python. The reposting script, integrated into the Production system, played a pivotal role in timely package tracking, resulting in a cost saving of 2 million dollars. Implementation of a Data Reconciliation and Data Quality (DRDQ) Engine in Scala ensured synchronization of data between source and target, maintaining consistency in terms of row count, data type, and metadata across various tables stored in PostgreSQL.
- Successfully managed the migration of data from one ScyllaDB Cluster to another using Spark. Employed PySpark for data transformation and Apache Spark (datastax spark-cassandra-connector Scala framework) for data extraction. The processed data, amounting to 21 TBs, was handled on a highly available distributed cluster with 2-factor node replication, showcasing a robust and scalable approach.
- Data Ingestion of 7TB from different local clients to Hadoop cluster, and ran spark jobs and python scripts for processing.

Education

Sep 2022 – Dec 2023 | **University: University of Montreal, Canada**
(MILA- Montreal Institute of Learning Algorithms)
*D*egree: Masters in Computer Science (Machine Learning Specialization), **Grade**: A
*C*ourses: Machine Learning, Data Science, Deep Learning, Natural Language Processing, Geometric Data Analysis

Aug 2013 – Jul 2017 | **University: Dr. A.P.J. Abdul Kalam Technical University, AKTU India**
College: Meerut Institute of Technology, MIET Group, Meerut.
Degree: B.Tech in Computer Science and Engineering, **Percentage**: 82.26/100 [Honours]

Selected Honors And Awards

2022, 2023 | **Diversity Award (Tuition Fees Scholarship/Exemption)**
Organization: University of Montreal

2019, 2020, 2021, 2022 | **Awards: IBM Eminence and Excellence Award, IBM Impact on the Business and Significant Achievement, IBM Service Excellence Award, IBM Growth Award, ScyllaDB Innovator Award**
Organizations: IBM India Pvt. Ltd, ScyllaDB (NoSQL Product Company)

2014, 2015, 2016 | **Awards: Academic Excellence Award, Codezilla Coding Award**
Organizations: Meerut Institute of Technology, Meerut Institute of Engineering & Technology, MIET Group

Skills: Soft and Technical Skills, Certifications, and Courses - Online Learning

Soft Skills | Leadership, Teamwork, Adaptability, Positivity, Interpersonal Skills, Creative thinking

Technical Skills | Tech- LLMs, RAG, Generative AI, Natural Language Processing (NLP), MLFlow, Distributed Machine Learning, VS Code, Azure ML, Hadoop, IBM Cloud, IBM Watson, Git, Docker, Kubernetes, Flask, Spark, Kafka, Cassandra, Microservices, Rest APIs, Jupyter Notebook, S3, Snowflake, Parameter-Efficient Fine-Tuning, object-oriented principles, Grafana, Prometheus, Kibana; Libraries - PyTorch, pandas, numpy, NLTK, Scikit-Learn, Streamlit, FastAPI, HuggingFace, CometML, Optuna, Ray, LangChain, Tensorflow
Programming Languages- Python, PySpark; Databases- NoSQL, MySQL, PostgreSQL, Elastic-search

Certifications and Online Courses | Microsoft Azure Fundamentals: 2021, IBM Certified Big Data Engineer: 2020, Enterprise Design Thinking Practitioner, IBM Watson Knowledge Catalog Essentials, Cognitive Practitioner (IBM), Python for Data Science (IBM), Deep learning specialization (Coursera)

Publications

2019 | **Optimal Partition Search**
A. Saxena and A. Saxena, "Optimal Partition Search," 2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), 2019, pp. 1-8, doi: 10.1109/ICECCT.2019.8869459. Link: https://ieeexplore.ieee.org/abstract/document/8869459

2017 | **DeepCoder: An Approach to Write Programs**
A. Saxena, A. Saxena, J. Patel, "DeepCoder: An Approach to Write Programs," 2017 International Conference on Advanced Research and Innovation in Engineering (ICARIE), 2017 International Journal of Engineering and Manufacturing Science (IJEMS), 2017, pp. 9-13, Vol. 7, No. 1, Research India Publications. Link: https://www.ripublication.com/ijems_spl/ijemsv7n1_02.pdf

Relevant Projects

End-to-End Large Language Model (LLM)-based Scalable Retrieval Augmented Generation (RAG)-powered Question-Answering (QA) App

- Application integrates a semantic embedding model to represent queries as vectors and utilizes a vector database for retrieving top-k relevant contexts.
- Employing a Eleuther AI's GPT-Neo 20B LLM, the project demonstrates fine-tuning in a multi-GPU cluster environment, showcasing data ingestion, context loading, fine-tuning, embedding, and indexing processes.
- App takes a new query, conducts vector similarity search, retrieves relevant contexts, and passes them, along with the query, to the LLM, which generates context-aware answers.
- Libraries and Techniques Used: Streamlit, Parameter-Efficient Fine-Tuning, Ray (for distributed LLM Fine-Tuning), Deta (To access Deta Vector Database), LangChain, PyTorch, FastAPI (To serve production-ready LLM App)

Question Answering

- Implemented complete pipeline having data cleaning, tokenization, initializing document store, retriever, reader, and evaluation.
- Compared BM25 and Dense Passage Retriever for Reader based on Recall Evaluation Metrics for top-[1, 3, 5, 10, 20] reader and reader. Results showed that Dense Passage Retriever boosted performance by 0.1 for top-[10,20], and saturated for top-[3, 5] retrievers and readers.
- Implemented domain adoptation and fine tune on SQUAD, SubjQA, SQUAD+SUBJQA datasets to compare the performance. Results showed 25% improvement on using the SQUAD+SUBJQA fine-tuned dataset.
- Evaluation and Exact Match/F1 score for Retriever Evaluation.
- Used Huggingface, Pytorch, and Haystack for transformer model building and evaluation.

Hockey NHL Project

- Downloaded the live hosted data using Python scripts and API. Cleaned the data according to the project requirements. Created the visualizations using contours, the intention is to showcase the shot generation intensity.
- Applied logistic regression, XGBoost, Neural Networks, Ensemble methods (Decision Trees, LGBM, Random Forests), Log models in Comet ML. Used Shapley and Lime to analyze the feature importance of the features.
- Developed flask API for prediction and download model from the registry. Created first docker containers for passing the data to generate prediction service (using Flask API) and second docker container for WEB UI through Streamlit. Establish the docker communication network.

Youtube Content GPT:- (Deployed App)

Used open-source google/flan-t5-xxl LLM hosted in Huggingface. Integrated Wikipedia tool with Huggingface LLM using LangChain and user interaction through the Streamlit app.

Climate Change: Spatio-temporal segmentation and tracking of weather patterns with light-weight Neural Networks

Experimented with CGNet model suitable for lightweight networks based on epochs, activation function, and different loss functions such as Cross Entropy and intersection Over Union. Results showed using contour plots and histograms.

Classifying Handwritten Digits (Modified MNIST), Text Classification

Implemented CNN Ensemble with voting classifier and logistic regression model. Implemented LSTM model, MLP Model, Naive Bayes. Implemented word2vec embeddings and explored pre-trained models and pre-trained embeddings.