# AARUSH GHOSH

+1(647) 229-7422 a66ghosh@uwaterloo.ca LinkedIn GitHub Portfolio

## EDUCATION

**University of Waterloo** ***Expected Apr 2028***
Bachelor of Mathematics: Statistics and Computer Science, Co-op

- **Relevant Courses:** Data Structures, Algorithms, Data-Intensive Distributed Computing, Networks and Distributed Computer Systems, Statistics, Optimization, Linear Modeling, Stochastic Processes

## SKILLS

**Languages:** Python, Go, SQL, TypeScript, C++
**Backend:** Apache Spark, Kafka, Flink, Airflow, DBT, Databricks, PostgreSQL, MongoDB, Redis, Neo4j, Parquet
**ML/AI & Cloud:** PyTorch, LangChain, LlamaIndex, TensorFlow, AWS, GCP, Azure, Docker, Kubernetes, Terraform
**Developer Tools:** Git, GitHub Actions, FastAPI, Django, Pytest, CI/CD, Linux, Pydantic

## EXPERIENCE

**Machine Learning Engineer (Incoming)** **Jan 2026 - Apr 2026**
Qorsa *Waterloo, Canada*

- Building **AI agents** and **RAG** applications for the AtlasAI platform using **PyTorch, LangChain, Neo4j**, focusing on NLP pipelines, **knowledge graph** construction, and LLM **fine-tuning** for quantum-secure enterprise solutions

**Data Engineer** **Jan 2025 - Apr 2025**
University of Waterloo: Dean of Mathematics *Waterloo, Canada*

- Automated budget planning for **9,000+ students** with **94.3% accuracy** using **Markov chain forecasting model** with **Python** deployed on **GCP**, enabling data-driven resource allocation for Dean of Mathematics
- Reduced data pipeline runtime **by 82%** by transforming legacy SQL queries into **PySpark** notebooks with automated scheduling, orchestration, monitoring via **Airflow**, **DBT** with **Apache Kafka** for data streaming
- Designed **asynchronous** logic for simultaneous ingestion from Oracle database and dynamic web data with robust backend in **Python**, **FastAPI** and **Redis**

**AI Engineer** **May 2024 - Aug 2024**
CGI *Toronto, Canada*

- Deployed production-grade **RAG** chatbot on **Databricks** processing Google Analytics 4 data through **medallion architecture**, enabling natural language queries over web traffic patterns for insurance client marketing team
- Built automated **ETL** pipelines from GA4 API using **Spark** to transform raw events into weekly aggregates
- Improved **text-to-SQL** agent accuracy by **23%** through schema **metadata enrichment** and implemented retrieval-augmented parsing with **LlamaIndex**, achieving **89% table cell F1-score** on complex multi-table marketing analytics queries

**Machine Learning Developer** **Sep 2023 - Apr 2024**
WAT.ai *Waterloo, Canada*

- Conducted research on unsupervised anomaly detection within Canada's largest undergraduate AI hub, implementing and comparing **K-means** and **DBSCAN** clustering algorithms on a **multi-class dataset** of 7 cyber-attack types across **105 IoT devices** to identify network intrusion patterns without labeled training data.
- Deployed an interactive research demo using **TensorFlow** and **TypeScript** for real-time browser-based visualization of clustering performance, enabling reproducible algorithm comparison.

## PROJECTS

**iRouter: SQL Query Engine Router** | *Python, Query Optimization, AST Parsing, Caching* **Nov 2025**

- Built **iRouter**, which auto-selects optimal backend (DuckDB, Polars, Spark) by **data size**, **query complexity**, and **metadata statistics** achieving **50x** performance gains over native SQL engines
- Implemented **partition pruning** functionality with **AST** based predicate extraction, reducing data scanned by **70-90%** on date-filtered queries across **Hive partitioned Parquet** tables
- Further optimized by implementing **LRU query cache** with TTL achieving **90%+** hit rates and latency improvements
- Developed **CLI** tool, automated test suite with **pytest** and integrated **GitHub Actions** for **CI/CD** pipeline

**Cascade: Distributed Task Orchestrator** | *Go, Concurrency, gRPC, etcd* **Sept 2025**

- Designed **Cascade** to coordinate **50+ workers** on long-running pipelines, achieving **exactly-once execution** and **8.5k+** tasks/sec throughput using **Raft** consensus via **etcd** for leader election and **distributed locking**
- Implemented **gRPC** task dispatch with **topological sort** for dependency resolution, **goroutine-based work stealing** with **automatic retry** and **exponential backoff**, reducing dispatch latency from **250ms to 12ms** while maintaining **92% worker utilization** on pipelines with **75+ task dependencies**