

Jacob Allan - Software, Machine Learning & Quantitative Systems

US/Canadian Citizen | jacob.allan@mail.utoronto.ca | (613)-532-2831 | [linkedin.com/in/jacob-allan-ml](https://www.linkedin.com/in/jacob-allan-ml) | github.com/Jakey794

EDUCATION

University of Toronto

Toronto, Ontario

BASc in Engineering Science (3.56/4.00 GPA)

Expected Graduation, May 2029

- **Concentrations:** Machine Intelligence, Mathematics | **Schulich Leader Scholar**
- **Relevant Coursework:** Data Structures & Algorithms, Probability & Statistics, Linear Algebra, Python/C

SKILLS

Languages: Python, Rust, C/C++, SQL, TypeScript/JavaScript, Java, MATLAB

ML/Data: PyTorch, scikit-learn, XGBoost, NumPy, Pandas, GMM/HMM, model evaluation, time-series validation

Software/Systems: Django, FastAPI, PostgreSQL, REST APIs, AWS, Terraform, Docker, GitHub Actions, pytest

Quantitative: Probability/statistics, GMM/HMM, VaR, stress testing, efficient-frontier optimization, portfolio P&L

EXPERIENCE

Northstar Downhole Specialists

Calgary, Alberta

Software Engineering Intern

Jun 2026 - Present

- Built the entire OdooRedo enterprise Documents/IPR system for operational workflows with Python 3.13, Django 5.2, PostgreSQL 17, RBAC, and audit trails.
- Implemented Documents/IPR workflows, RCA/CAPA, approvals, notifications, external submissions, enterprise access controls, and rollback-safe Odoo migration/reconciliation.
- Provisioned AWS across ECS, RDS, S3, SES, ALB, and WAF with Terraform; built nine-job CI spanning integration, E2E, typing, containers, and security.
- Validated a synthetic 40-user, 200-request workload with zero unexpected errors and ~128 ms p95 latency; CI covers 1,286 PostgreSQL-backed tests and 13 Playwright tests.

Royal Military College of Canada

Kingston, Ontario

Machine Learning Researcher

May - Sept 2025

- Built and trained 5+ PyTorch CNNs for RF spectrogram classification using custom Python and RTL-SDR pipelines.
- Curated a 150K-sample, 5-class RF dataset and achieved ~90% average accuracy on unseen real-world signals.
- Improved noisy-band generalization by ~10% through data augmentation, learning-rate decay, frequency-offset sampling, and spectrogram normalization.

University of Toronto Engineering Finance Association

Toronto, Ontario

Portfolio Manager - promoted from Sales & Trading Analyst

Sept 2025 - Present

- Built a 22-factor XGBoost return-prediction model for a 50-participant quant contest using macro, rates, momentum, and volatility features.
- Promoted to Portfolio Manager after applying ML-based prediction, optimization, and risk modeling.
- Apply VaR and efficient-frontier optimization to portfolio risk, asset allocation, and thesis-driven trade proposals.

PROJECTS

LLM Reliability + EvalOps Platform | FastAPI, PostgreSQL, Next.js, Docker, GitHub Actions |

Aug 2026

- Built a full-stack platform for versioned datasets, prompts, and models; deterministic and LLM-based graders; evaluation dashboards; and CI quality gates.
- Caught a controlled 20-case RAG regression: pass rate fell from 95% to 85%, failures rose from 1 to 3, and estimated cost more than doubled.
- Shipped CI with 221 backend tests, a frontend build, and an eval gate enforcing pass-rate, score, cost, and p95 latency.

Market Regime + Portfolio Risk Platform | Python, scikit-learn, GMM/HMM, Streamlit |

Aug 2026

- Built a leakage-aware Python platform for ETF ingestion, portfolio-risk metrics, GMM/HMM/KMeans regime models, stress tests, and cost-aware backtests.
- Prevented look-ahead bias with chronological splits, train-only scaling, shifted signals, and future-mutation tests; shipped a 322-test CI matrix across Python 3.12-3.14, research memos, and a nine-page dashboard.

Event-Driven Trading Engine | Rust, Python, Criterion, GitHub Actions |

Aug 2026

- Built a Rust matching engine with price-time priority, partial fills, deterministic replay, portfolio P&L, pre-trade limits, and a kill switch; CI passes 247 tests.
- Benchmarked the 10K-event core path at 125 ns p50 and ~4.6M events/s on an Apple M4 Pro versus ~1.2M events/s for a naive Python baseline.