

Bruce Brasseur

Quantitative Developer

Beaufort, SC | (843) 476-7032 | brucebrasseur@gmail.com

Portfolio: <https://brasseur.dev>

GitHub: <https://github.com/BruceBrasseur> LinkedIn: <https://linkedin.com/in/bruce-brasseur>

PROFESSIONAL SUMMARY

Quantitative Developer with hands-on experience across deterministic backtesting, strategy simulation, market-impact and order book modeling, cyclic arbitrage scanning, and BTC time-series anomaly detection. Strong Python and Rust background across algorithmic trading, market data workflows, statistical analysis, optimization, machine learning, and reproducible research. Built trading software spanning equities, ETFs, and crypto with execution modeling, fees, slippage, borrow / funding costs, and analytics including drawdown, rolling Sharpe, exposure, turnover, and trade blotter reporting.

CORE SKILLS

Languages: Python, Rust, SQL, SQL Server, C++, JavaScript, TypeScript, R, MATLAB

Quant / Trading: quant trading, quantitative developer, algorithmic trading, backtesting, strategy research, strategy simulation, market data, equities, ETFs, crypto, market microstructure, order book simulation, execution modeling, pairs trading, mean reversion, breakout strategies, arbitrage scanning, time series analysis, Monte Carlo simulation, optimization, statistical analysis, reproducibility, deterministic systems

ML / Analytics: PyTorch, TensorFlow, Keras, scikit-learn, NumPy, Pandas, SciPy, matplotlib, anomaly detection, LSTM, GANs, machine learning, reinforcement learning, world models, model evaluation

Platform / Infra: Linux, FastAPI, REST APIs, Docker, Next.js, React, Axum, Tokio, Reqwest, Serde, Chrono, Git, Playwright, Caddy

SELECTED QUANT / TRADING PROJECTS

rust-backtester — Rust, Axum, Tokio, Next.js, TypeScript, TradingView Lightweight Charts, Recharts, Polygon.io

- Built a deterministic backtesting platform for equities, ETFs, and crypto with single-asset and pairs strategies, configurable fees, slippage, borrow / funding costs, and strict signal-on-close / fill-on-next-open execution semantics.
- Implemented MA crossover with optional volatility targeting, z-score mean reversion, Donchian breakout, and pairs z-score strategies using rolling ratio or OLS beta hedge.
- Stored run configs as JSON for reproducibility and exposed analytics including equity curve, drawdown, rolling Sharpe, exposure, turnover, returns histograms, and sortable trade blotter.

market-impact-sim — Python, order book simulation, market microstructure

- Built a from-scratch price-time-priority matching engine with max-heap bids and min-heap asks to measure market impact from a large order against heterogeneous noise and value traders.
- Simulated 700 noise agents and 100 value agents, generated baseline vs. impact runs plus order-volume sweeps from 500 to 50,000 units, and observed displacement growth roughly consistent with the square-root impact law.

HopScout — Python, Web3, multicall, NetworkX, DeFi, UniswapV2

- Built a cyclic arbitrage scanner for UniswapV2-style decentralized exchanges that discovers pools, batches reserve queries, models liquidity as a multigraph, enumerates 3-4 hop cycles, simulates swap paths with fees, and uses ternary search to maximize profit.

BTC-GAN-anomaly-detection — Python, TensorFlow, Keras, LSTM, Pandas, scikit-learn

- Built a time-series anomaly detection pipeline on hourly BTC/USD OHLC data using an autoencoder-compressed latent space, rolling windows, adversarial training, quantile-based thresholds, and clustered anomaly events for interpretable signals.

EvoLoss — Python, PyTorch, DEAP, genetic programming, CLI development

- Built a system that evolves differentiable loss functions for neural networks using expression-tree search, tournament selection, crossover, mutation, deterministic seeding, and artifact logging for reproducible experimentation.

PROFESSIONAL EXPERIENCE

National Science Foundation Funded Undergraduate Researcher — University of South Carolina Center of Computational Robotics | May 2021 - Aug 2021

- Conducted 50+ large-scale simulations and used 100+ compute hours on a high-performance computing cluster to evaluate discrete vs. continuous world-model methods, with heavy emphasis on reproducible experiment workflows, model evaluation, and comparative analysis.
- Built and modified Python, TensorFlow, and PyTorch research code for simulation, hyperparameter search, logging, and repeated-rollout analysis; contributed results to a research poster and paper.

Database Developer (Student Volunteer) — The Children's Center Hilton Head | Nov 2020 - Jun 2021

- Built 20+ dynamic SQL Server queries and reusable reporting workflows for 100+ students, joining data across multiple tables and iterating dozens of times with stakeholders to refine outputs, data validation, and data integrity.
- Replaced a manual paper-based process with weekly query-driven outputs used for report cards, performance analysis, and operational decision-making.

EDUCATION & CREDENTIALS

B.S., Computer Science, University of South Carolina Beaufort, Dec 2025 | Minor: Mathematics

Deep Learning Specialization (DeepLearning.AI)

UofSC Magellan Mini-Grant recipient | Provisional patent on an algorithm for evolving machine learning cost functions | Accepted into the LISD conference at Oxford for work on evolved evaluation functions