Braedyn Thompson

(949) 565-5903 • braedynthompson@berkeley.edu • Berkeley, CA | github.com/braaaeeedyn | braedynthompson.com

EDUCATION

University of California, Berkeley **Graduation: Fall 2027**
Bachelor of Arts in Computer Science, Bachelor of Arts in Data Science **GPA: 3.75/4.00**

WORK EXPERIENCE

Data Science & Machine Learning Research Intern — September 2026 – Present

Lawrence Berkeley National Laboratory Berkeley, CA

- Outage gaps in ~4 years of daily XCache traffic could leak into any of 800+ model configurations across 23 research runs; built the shared Pandas data path behind every experiment, with causal forward-fill of 37 outage days masked from every metric, chronological 70/15/15 splits, and cached, resumable GPU sweeps that skip scored configs, so any result re-materializes from its hash and seed alone.
- Next-day XCache hit volume had a persistence baseline to beat, and an ensemble only counted if it won on unseen folds; picked fixed weight PatchTST blends (PyTorch, neuralforecast) by leave-one-fold-out over 8 rolling 100-day folds against a pre-registered bar, and the adopted blend landed 25% below persistence on test and 9–16% below prior papers' LSTM, CNN-LSTM and Chronos-2 models.
- Adding 33,576 hourly rows to the daily XCache pipeline opened a new path for test data to leak; built shift-aligned hourly-to-daily joins with leakage assertions and hash-froze code and results, so every hour reconciles to its daily total and each accepted run passed 566 checks with zero leakage. The hourly feed also drives a same-day re-forecast that cuts test MAE from 15.27 to 3.10 TB/day by 18:00.

Machine Learning Research Intern — May 2026 – Present

California Baptist University Berkeley, CA

- An inverse-FEA predictor had stalled at $R^2 \approx 0.10$ ahead of the team's IEEE submission, and the open question was whether more model tuning could move it; benchmarked 7 models (Random Forest, XGBoost, HGB/ExtraTrees, PCA pipelines) and ran a paired clean-vs-noisy noise-identifiability sweep, which traced the ceiling to the data and moved the team off model tuning.
- SeismicSoCal's detector had to train on data cut exactly like the live SeedLink stream, from a USGS catalogue that caps each query at 20K rows; built a cached ingest of ~424K events whose monthly chunks split automatically at the cap, stored response-corrected 100 Hz waveforms with per-trace scales (~2 GB), and put one station list and preprocessing path behind training and live, which exposed a shortcut the detector had learned (response-corrected quakes vs raw-count noise) that a shared 1 Hz high-pass removed.
- Earthquake alerts are useless if late or false, so SeismicSoCal had to prove itself on unseen days before alerting anyone; built a 19-station SeedLink pipeline with multi-station location, a negative-evidence check and two-stage FCM pushes, kept the feed alive with a daemon the API server respawns when stream drops ($0 Oracle Cloud VM), and ran in shadow mode while a nightly job scored live log against USGS. All 145 push alerts across 80 held-out replay days were real quakes, 92% within 60 km, with the first alert ~25–31 s after origin.

Software Engineer Intern — September 2025 – June 2026

NumIsToken Berkeley, CA

- Each redemption action took ~2–3 s because the backend lifecycle APIs repeated database lookups and validation checks; rewrote the queries to drop the redundant work, then containerized and deployed the services with Docker. Load times fell ~85%, to ~300 ms.

PROJECTS

BayRidership | *PySpark, BigQuery, Dagster, LangGraph, LightGBM, Terraform, dbt, GCS* — August 2025 – October 2026

- BART publishes 67.8M hourly origin-destination rows and Bay Wheels adds 26M trips in monthly zips with two schemas; the warehouse had to take both at $0/month and account for every dropped row. Cleaned BART in PySpark and Bay Wheels in DuckDB, then modeled a BigQuery star schema in dbt with incremental facts and an SCD2 station snapshot, run by partitioned Dagster assets on an Oracle VM.
- BART and Bay Wheels publish on different schedules, and everything downstream had to stay current; set Dagster to reload BART within 6 hours of a change, rebuild dbt after ingest and retrain forecast weekly, with each KPI defined once in a CI-tested metrics spec.
- With no ops team, every push to main had to test and deploy itself without stored cloud keys; wrote Terraform for all 43 GCP resources, set up keyless GitHub Actions deploys through Workload Identity Federation, and tested a 119 MB non-root image before each release, behind 306 pytest, node and Playwright tests in CI. Locust load tests hold ~30 req/s at 25 users with a forecast p95 of 14 ms.

UC Campus Rent | *Airflow, PostgreSQL, dbt, Great Expectations, Terraform, BigQuery, GCS* — Jan 2026 – Present

- Rent data around UC campuses lives in seven public sources at different grains and cadences, and three publishers block or forbid scripted downloads; the goal was one monthly table per campus ring (30 ZIPs within 5 km of 4 UC campuses) that no bad file could corrupt. Gave each source an Airflow DAG (download, Great Expectations check, load) that isolates failing files, loaded idempotently into a Postgres raw layer tracked by a manifest, read blocked sources from browser downloads instead of scraping, and modeled it all in dbt (20 models, 61 tests, snapshot of Redfin's revisions). A failing source test stops run before marts, which keep their last good build.
- Worked under a $0-a-month limit on a free Oracle VM shared with other live services; capped Airflow and MLflow at 3 GiB total with systemd MemoryMax, ran every DAG at 13:00 Pacific outside the VM's busy hours, published the marts nightly to a BigQuery serving dataset and as JSON for the site, archived raw files in GCS, and wrote the GCP side (bucket, dataset, service account, Workload Identity Federation, $1 budget alert) in Terraform, with CI running lint, dbt parse, 227 unit and DAG tests and shellcheck.

TECHNICAL SKILLS

Python, SQL, Java, PySpark, BigQuery, PostgreSQL, DuckDB, dbt, Airflow, Dagster, Great Expectations, ETL/ELT, Data Warehousing, Data Modeling, Star Schema, SCD2 & Snapshots, Incremental Models, Partitions & Sensors, Data Quality Testing, Data Lineage, Monitoring & Alerting, Terraform, GCP (GCS, Cloud Run, IAM), Oracle Cloud, Docker, systemd, GitHub Actions CI/CD, MLflow, Pandas, pytest, Git