

Xinhuiyu (Huiyu) Liu

Sunnyvale, CA | xinhuiyuliu@gmail.com | (434) 242-9031

linkedin.com/in/xinhuiyu-huiyu-liu | github.com/liuxhy | Authorized to work in the U.S. without sponsorship

Education

Georgia Institute of Technology, M.S. in Computer Science

Expected Dec 2026

- **Coursework:** Graduate Algorithms, Operating Systems, Software Development Process, Machine Learning, Deep Learning

University of Virginia, Ph.D. in Environmental Sciences

Aug 2019 – May 2025

Experience

Dyneti Technologies (fintech startup)

Jun 2026 – Jul 2026

Software Engineer

San Mateo, CA

- Designed, developed, tested, and deployed an asynchronous fraud-detection workflow for a credit-card scanning **mobile SDK**, cutting scan-to-response latency by over **75%** by decoupling server-side ML inference from the synchronous scan path
- Built a **server-to-server Fraud Decision API** allowing merchant backends to retrieve fraud results precomputed at scan time, eliminating a redundant second scan and query-time inference delay
- Defined the async lifecycle across a **Java/Kotlin** Android SDK and **Node.js** backend, covering API contracts, RDS-backed schemas, inference state transitions, and pending-result handling, while preserving backward compatibility for live partner integrations
- Owned releases end to end via **GitHub Actions CI/CD**, pairing automated unit/integration tests with multi-device **instrumented tests** covering async workflows, failure handling, and edge cases; served **on-call** rotations, debugging production issues via **CloudWatch** logs and **Grafana** dashboards

IBSS Corp

Sep 2025 – May 2026

Software Engineer Intern

Remote, US

- Built **AWS Glue, Spark, and Sedona** pipelines to ingest, validate, transform, and spatially align four heterogeneous raster and vector datasets (forecast, observation, terrain, and land-water masks) into standardized 3 km model inputs
- Designed the data layer around access patterns: **PostgreSQL/PostGIS** for relational and spatial metadata, **TimescaleDB** for **10M+ time-series points**, and **AWS S3** for **TB-scale gridded outputs**, enabling queries with **p95 latency under 200 ms**
- Deployed containerized **microservices** on AWS, using **Lambda** to trigger hourly event-driven ingestion and GPU-based **CorrDiff** inference for 3 km to 1 km downscaling, while isolating compute-intensive workloads from latency-sensitive services
- Served downscaled forecasts through **FastAPI REST APIs** and interactive **React/TypeScript and Leaflet** maps, supporting automated frost and heat alerts with **87% accuracy**

TGS (leading energy data analytics company)

May 2024 – Aug 2024

Data Engineer Intern

Houston, TX

- Built scalable geospatial **ETL pipelines** on GCP using Python, Xarray, and Dask to ingest, validate, and harmonize **20 TB** of gridded WRF outputs and offshore lidar observations across spatial, temporal, and vertical dimensions
- Designed cloud-optimized storage layouts for multidimensional raster and tabular data using **Zarr and Parquet**, **reducing processing time 70%** through optimized chunking and partitioning while supporting distributed, memory-efficient analytics
- Developed reusable spatial interpolation and data-fusion workflows using PyKriging and a gradient boosting model to produce analysis-ready offshore wind datasets, **reducing wind-speed RMSE 15%**

University of Virginia

Aug 2019 – May 2025

PhD Researcher – Climate Data Engineering

Charlottesville, VA

- Built reusable parallel **Python/Xarray/Dask** pipelines to ingest, quality-control, harmonize, and extract features from **100+ TB** of satellite, reanalysis, and climate-model data on multi-node **Slurm HPC clusters**
- Packaged the pipelines as documented, version-controlled Python modules adopted by **2** research groups for their own datasets
- Authored **3 first-author publications** in leading climate journals, presented at **7+ international conferences**, and received the competitive **NASA FINESST Fellowship** and **Best Student Presentation** at the American Meteorological Society Annual Meeting

Projects

Synchronized Distributed File System | C++, gRPC, Protobuf, inotify, pthreads

- Built an AFS-style distributed file system with **whole-file client caching** under **single-writer semantics**, using CRC32 checksums paired with mtime comparison to skip redundant transfers and **last-writer-wins** resolution on conflicts; streamed files in chunks with gRPC deadlines for graceful timeout handling on 100 MB+ transfers
- Synchronized clients in real time via **inotify watcher thread** and **persistent async gRPC callback stream** pushing file and deletion updates instead of polling; enforced lock ownership on every write, with cleanup restoring server state after mid-write client crashes

Skills

Languages: Python, Java, Kotlin, JavaScript/TypeScript, SQL, C++

Systems & Distributed Computing: Spark/PySpark, Sedona, Dask, gRPC, Protobuf, concurrency, Docker, Unix/Linux

Web & Mobile Development: Node.js, Express, FastAPI, React, Android SDK, REST APIs

Cloud & Tooling: AWS (S3, Glue, Lambda, RDS, CloudWatch, IAM), GCP, GitHub Actions CI/CD, Git

Geospatial, Data & ML: raster/vector data, PostGIS, PostgreSQL, TimescaleDB, Xarray, NetCDF, Zarr, Parquet, GeoTIFF/COG, GDAL, Leaflet, GeoJSON, PyTorch, scikit-learn, QGIS