Daniel David | Portfolio

Software Engineer | Specialized in Scalable & Secure ML Pipelines | New York, NY | [LinkedIn] | [GitHub]

---

1. Tutorial #1: Basic Usage of Rhino FCP for Federated Learning

During my tenure at Rhino Health, I designed and authored the flagship technical tutorial for the Rhino Federated Computing Platform (FCP). This project served as the primary onboarding resource, meticulously guiding users through critical platform operations such as project initialization, importing datasets from cloud-mounted storage, and executing containerized code for training and inference. By standardizing these core fundamental steps, the tutorial significantly improved user autonomy and streamlined the adoption of distributed ML workflows across the platform's diverse client base. Link. Link.

2. Tutorial #2: Using the Rhino Health Python SDK

In addition to core platform guides, I wrote and maintained the specialized technical tutorial for the Rhino Health Python SDK. This project focused on empowering developers to interact programmatically with the Federated Computing Platform, covering advanced topics such as automated data schema validation, programmatic project management, and the integration of local development environments with cloud-based federated workloads. This resource remains a critical tool for engineers looking to build scalable, scriptable pipelines within the Rhino ecosystem. Link. Link.

3. Model & Code Encryption for Secure Federated Computing

To bolster the security of Rhino's Federated Computing Platform, I engineered a robust workflow for the encryption of machine learning models and training logic. This implementation leveraged a dual AES+RSA encryption strategy, integrated with AWS Secrets Manager or JSON keyfiles for high-integrity key management. Alongside the development, I authored a comprehensive technical article that serves as an advanced reference guide, enabling researchers and enterprise clients to perform secure parameter aggregation without compromising sensitive intellectual property. Link. Link.

4. Linear Algebraic Approaches to Neuroimaging Data Compression

In this academic project, I performed a comparative analysis of linear algebraic methods for the efficient compression of high-dimensional neuroimaging data. I implemented both Matrix (SVD) and Tensor (Tucker) Decomposition techniques to evaluate their effectiveness in reducing data dimensionality while retaining essential biological features. My findings highlighted the superior performance of tensor-based decomposition in medical imaging contexts, providing a pathway for significant reductions in storage and computational overhead for large-scale clinical datasets. Link. Link.

5. Federated Learning under Temporal Data Drift

This research project explored the resilience of federated learning systems when subjected to temporal data drift, a real-life realistic phenomenon where client data distributions shift over time. Utilizing the Flower framework and PyTorch, I conducted simulations using datasets like Fashion-MNIST to evaluate how global models adapt to these evolving patterns. My work specifically addressed the risk of "categorical forgetting" in decentralized nodes, leading to the development of training strategies that balance model adaptation with long-term stability in dynamic, real-world environments. Link. Link.

6. Data Analyst Career Explorer: Market Trends Analysis

In this data-driven exploration, I executed an end-to-end analysis of the 2023 data analyst job market using PostgreSQL and real-world posting data. I was responsible for designing the database schema and optimizing complex SQL queries to extract insights on top-paying roles, critical technical skills, and industry growth patterns. The final project provided a structured look at market demand, demonstrating my proficiency in data engineering, schema optimization, and the ability to turn unstructured data into strategic career insights. Link.