Hydropower Intelligence Platform: Real-time and Historical Capacity Factor Analytics using Microsoft Fabric

1. Introduction

This project presents a comprehensive data engineering and analytics workflow built using Microsoft Fabric for hydropower analysis. It combines real-time data from Fingrid's open API with historical hydrological data from Zenodo and the JRC Hydro Power database. The objective was to design a unified lakehouse architecture for analyzing capacity factor deviations, integrating semantic search, and demonstrating Retrieval-Augmented Generation (RAG) for intelligent querying of time-series insights.

2. Lakehouse Architecture

The solution follows the Bronze–Silver–Gold layered design pattern in Fabric, using Apache Spark and Delta tables for scalable processing. This approach ensures data consistency, reusability, and clear transformation lineage.

Layer	Purpose	Data Sources
Bronze	Raw ingestion zone storing unprocessed data.	Fingrid API (3-min intervals), Zenodo inflow dataset, JRC Hydropower metadata
Silver	Cleaned and harmonized analytical dataset.	Aggregated hourly data joined with historical baselines and plant metadata
Gold	Curated dataset for analytics and visualization.	Derived KPIs such as observed_cf, baseline_cf, and deviation_cf

3. Analytical Workflow

Hydropower production data was aggregated from 3-minute to hourly intervals. Historical baselines were computed from Zenodo's capacity factor dataset to calculate deviations between real-time and historical performance. The gold tables contained hourly KPIs such as observed capacity factor, baseline average, and deviations. Power BI visualizations were designed to show hourly and monthly deviations, along with inflow and generation trends.

4. Key Analytical Insights

- October 2025 data showed lower-than-baseline production due to seasonal inflow variation.
- Run-of-river (HROR) and dam-based (HDAM) plants displayed distinct deviation patterns.
- The methodology successfully integrated real-time and historical hydropower performance analytics.

5. Retrieval-Augmented Generation (RAG) Implementation

To enable semantic exploration, each hydropower observation was converted into a textual summary describing system performance. These summaries were embedded using Azure OpenAI's text-embedding-3-small model and indexed using FAISS for vector-based retrieval. This setup allowed similarity searches such as identifying hours with comparable deviations or hydrological conditions.

However, due to region restrictions in Microsoft AI Foundry (only US East 2 and Sweden Central supported), deployment of the chat completion model for answer synthesis was not possible in Norway East. Thus, the agentic component of the RAG pipeline remained incomplete, though retrieval and embedding processes were validated successfully.

6. Challenges and Learnings

1. Regional compliance policies prevented AI model deployment in allowed zones.
2. Data schema conflicts (integer vs long) caused Delta write inconsistencies, requiring explicit type casting.
3. Index alignment between Spark and FAISS required ordered indexing rather than Spark's non-sequential IDs.
4. Despite these issues, the workflow successfully demonstrated embedding generation and vector retrieval capabilities.

7. Future Scope

- Enable Chat Completion models in Norway East once available in AI Foundry.
- Extend Fingrid data ingestion to multiple years for stronger seasonal analysis.
- Integrate retrieval results into a Power BI Q&A dashboard for contextual hydropower search.
- Deploy a full RAG pipeline for anomaly explanation and automated report generation.

8. Conclusion

The project established a robust foundation for integrating real-time and historical hydropower data in Microsoft Fabric. While the agentic RAG feature could not be deployed due to platform constraints, the implementation demonstrated the potential for AI-enhanced data analytics in the energy sector. This academic exercise provides a blueprint for future smart data platforms in renewable energy monitoring.