D607 TASK 1

DATA ANALYTICS AT SCALE- D609

5/25/2025
WESTERN GOVERNORS' UNIVERSITY
Zainab Abbas

Big Data Architecture Proposal: STEDI Human Balance Analytics

Executive Summary

The purpose of this proposal is to design and implement a robust and scalable big data lakehouse architecture that supports STEDI's Human Balance Analytics project. As a data engineer for STEDI, I am responsible for creating a data infrastructure that ingests, processes, and curates sensor data collected from the STEDI Step Trainer devices and companion mobile applications. This infrastructure will enable the data science team to develop accurate machine learning models aimed at detecting steps and analyzing human balance in real time (Udacity, n.d.).

This project is expected to yield several key outcomes. By developing a comprehensive data pipeline that respects user privacy and filters data based on consent, STEDI can ensure ethical data handling while enabling high-quality machine learning model development. The architecture will support scalability for future device rollouts and provide efficient access to data for analysis, ultimately improving product performance, supporting user trust, and positioning STEDI as a leader in health-tech innovation (Amazon Web Services, n.d.-a).

Business Problem Recap

STEDI has launched a wearable Step Trainer and mobile application that together capture detailed motion data to help users improve their balance. However, the organization faces several challenges in handling this data. A defect in the STEDI fulfillment website led to repeated use of the same 30 serial numbers for millions of customers. This has caused ambiguity in matching sensor data with the correct user. Additionally, not all customers have agreed to share their data for research purposes, making it critical to ensure that only authorized records are used for machine learning. Without a structured, scalable data pipeline to address these issues, the effectiveness of STEDI's analytics and model training is severely compromised (Udacity, n.d.).

Section 2: Needs Assessment

Big data analytics offers a viable solution to the challenges STEDI faces by enabling the processing of high-volume, semi-structured data in a distributed environment. Using tools such as AWS Glue and Apache Spark, I can build ETL (Extract, Transform, Load) pipelines that filter, clean, and integrate sensor and customer data efficiently. These tools support the automation of complex data workflows and facilitate real-time insights, which are essential for creating effective machine learning models (Zaharia et al., 2016; Amazon Web Services, n.d.-a).

The methods I will use include distributed batch processing via PySpark to manage data transformations and aggregations, SQL queries through Amazon Athena for validation and exploration, and schema inference using Glue Crawlers to dynamically adjust to evolving data structures. These approaches are specifically suited for managing large datasets that vary in structure and volume, such as those generated by IoT devices and mobile applications (Amazon Web Services, n.d.-b).

Section 3: Solution Design

To support this analytics solution, the architecture will leverage several AWS services. Amazon S3 will serve as the foundational data lake, hosting data across three zones: landing, trusted, and curated. AWS Glue will manage ETL jobs, using PySpark to clean and transform the data. AWS Athena will provide serverless querying capabilities, enabling analysts to explore and validate data stored in S3 (Amazon Web Services, n.d.-b).

The proposed data model includes a structured zoning approach. In the Landing Zone, raw JSON files are ingested directly from the customer, step trainer, and accelerometer sources. The Trusted Zone contains data that has been filtered based on user consent to share data for research. Finally, the Curated Zone includes joined and enriched datasets prepared specifically for machine learning training.

An integrated solution will be built by combining Glue jobs for data processing, S3 for storage, Athena for querying, and potentially QuickSight for visualization. The Glue jobs will perform the necessary filtering and joining operations, ensuring that only customers with shared consent and valid sensor readings are included in the training datasets. SQL queries in Glue will be used extensively for their reliability and consistent output (Amazon Web Services, n.d.-a).

To manage dependencies and automation, Glue Workflows will be used for job orchestration. IAM roles and bucket policies will enforce data security and privacy. All code, including SQL DDL scripts and Python Glue jobs, will be stored and versioned using GitHub to ensure reproducibility and collaboration.

Justification of Choices

AWS Glue and Spark are the optimal choices for this project due to their ability to scale dynamically and process large, semi-structured data efficiently. Their integration with S3 and Athena creates a seamless pipeline from data ingestion to insight generation. Athena provides an interactive SQL interface without requiring infrastructure provisioning, which is ideal for validating outputs at each stage (Amazon Web Services, n.d.-b).

These architectural choices align directly with STEDI’s business needs. They enable data scientists to train models using clean, curated datasets, which leads to more accurate predictions. They also ensure that data governance is enforced through consent filtering and secure data access. These solutions are cost-effective and scalable, making them suitable for STEDI’s anticipated growth and user expansion (Zaharia et al., 2016).

Future Enhancements

Future iterations of the architecture can include real-time data ingestion through Amazon Kinesis, which would allow immediate feedback on user balance. Amazon SageMaker could be used to automate model training and deployment. Additionally, visualization tools such as Amazon QuickSight can be integrated for real-time dashboards. Data validation services could be incorporated to monitor schema consistency and catch anomalies early in the pipeline (Amazon Web Services, n.d.-a).

Implementation Plan

To implement this architecture, I will begin by uploading the provided JSON datasets to their respective S3 landing zones. Using AWS Glue Studio, I will create Glue tables for the customer, accelerometer, and step trainer landing zones and manually define schemas using SQL DDL scripts. Glue jobs will be written in PySpark to filter and sanitize data, creating trusted tables. Further jobs will join and aggregate trusted data to generate curated tables. Each stage will be validated in Athena with record count screenshots and saved for submission. All code will be documented in a GitHub repository, and a PDF with screenshots will be submitted separately from this paper (Udacity, n.d.; Amazon Web Services, n.d.-a).

References

Amazon Web Services. (n.d.-a). *What is AWS Glue?* Retrieved from https://docs.aws.amazon.com/glue

Amazon Web Services. (n.d.-b). *Build a Data Lake on AWS*. Retrieved from https://aws.amazon.com/big-data/datalakes-and-analytics

Amazon Web Services. (n.d.-c). *AWS Athena Documentation*. Retrieved from https://docs.aws.amazon.com/athena

Udacity. (n.d.). STEDI Human Balance Analytics Project Resources. https://github.com/udacity/nd027-Data-Engineering-Data-Lakes-AWS-Exercises

Zaharia, M., Chowdhury, M., Franklin, M. J., Shenker, S., & Stoica, I. (2016). Apache Spark: The Definitive Guide. O’Reilly Media.