

Software Requirements Specification (SRS)

Project: Smart Manufacturing: Fuel Efficiency and Turbine Health Analytics on Azure Cloud

1. Introduction

1.1 Purpose

This document defines the software requirements for the Turbine Health & Fuel Efficiency Analytics Platform. The goal is to collect, process, and analyze turbine sensor data to monitor fuel efficiency, detect turbine health degradation, and provide actionable insights through FAST APIs and dashboards.

1.2 Scope

The platform will ingest raw turbine sensor data, clean and transform it, store it in structured layers (Raw, Silver, Gold), compute KPIs, detect anomalies, and provide insights via REST APIs and Databricks dashboards for operators and engineers.

1.3 Definitions, Acronyms, and Abbreviations

ETL – Extract, Transform, Load

ADLS – Azure Data Lake Storage

ADF – Azure Data Factory

KPI – Key Performance Indicator

API – Application Programming Interface

2. Overall Description

2.1 Product Perspective

The system acts as a middleware layer between turbine sensors and end-users. It ingests raw telemetry data, processes it via ETL, stores structured data in Azure Data Lake, and provides analytics through APIs and dashboards.

2.2 Product Functions

- Data ingestion from turbine sensors or files
- Data cleaning and transformation
- KPI computation (fuel efficiency, decay coefficients, anomaly detection)

- FAST API for turbine health and performance data
- Visualization dashboards using Databricks

2.3 User Classes and Characteristics

- Operator: Non-technical, monitors dashboards and alerts
- Data Scientist: Performs analysis and modeling
- API Client: Consumes FAST APIs
- Administrator: Manages pipelines, storage, and APIs

2.4 Operating Environment

Cloud environment (Azure Data Lake, Data Factory, Databricks) with local ingestion scripts. Application layer developed in Python using FastAPI.

2.6 Assumptions and Dependencies

- Continuous sensor data in correct format
- Availability of Azure services
- Reliable network connectivity

3. System Features

FR1: Ingest raw turbine data

FR2: Validate column structure

FR3: Clean and transform data

FR4: Compute KPIs

FR5: Detect anomalies

FR6: Store data in Raw/Silver/Gold layers

FR7: Provide FAST APIs (health-summary, sensor-metrics, anomaly-alerts)

FR8: Deliver dashboards via Databricks

4. External Interface Requirements

- Swagger UI for API documentation
- Databricks dashboards for visualization
- Azure Data Lake and Data Factory integration

5. Non-Functional Requirements

- Performance
- Scalability
- Reliability & Availability
- Maintainability
- Security: HTTPS APIs, encrypted storage
- Usability: Simple dashboards for operators