

Final Project Proposal



School of Advanced Technology

Algonquin College of Applied Arts and Technology

25W_CST2213_300: Business Intelligence Programming 2

Professor Siju Philip

March 08, 2025

Written by:

Vasanth Gnana Seelan (041112627)

AI-Powered API Consumer Analytics & Anomaly Detection for Financial Services

(An On-Premise Data Engineering & BI Solution for API Usage Insights)

Problem Statement/Objective:

Financial institutions provide APIs for services such as transaction processing, fraud detection, market data retrieval, and customer verification. However, the lack of detailed analytics makes it difficult to understand API consumer behavior, detect anomalies, and optimize API performance.

This project aims to develop an AI-powered API analytics dashboard that provides deep insights into API usage trends, consumer behaviors, system performance, and security threats. The solution will be fully on-premise, leveraging data pipelines, advanced analytics, and machine learning for anomaly detection.

Objectives and Goals:

Primary Objective:

Develop a next-gen API analytics and anomaly detection system to provide real-time insights into API consumer behavior, performance, security trends, and fraud detection.

Key Goals:

- Track API usage patterns – Requests per consumer, endpoint popularity, and response times.
- Detect anomalies & security threats – Unusual API spikes, repeated failed authentications.
- Optimize API performance – Identify slow endpoints and high-latency issues.
- Build an interactive Power BI dashboard – Real-time API analytics & security insights.
- Use AI/ML for fraud detection – Identify suspicious API calls & automate API monitoring.

Data Requirements and Requirements Gathering:

A list of required data sources:

- API Gateway Logs (e.g., Nginx, Apache, Kong)
- Direct API Call Tracking (e.g., Postman, internal request logs)
- Transaction Logs from financial APIs
- Error Logs (authentication failures, rate-limiting violations)
- System Performance Metrics (CPU, memory, latency tracking)

Data Specifications:

- Formats: JSON, CSV, Log Files, SQL Tables
- Types of Variables:
 - Categorical: API consumer types, endpoint names, error categories
 - Numerical: Request counts, response times, failure rates, memory usage
 - Temporal: Timestamps (standardized to UTC)
- Time Frames:
 - Historical data for trend analysis (e.g., past 12 months)
 - Real-time or near real-time processing for anomaly detection

Data Collection and Data Sources / Scope Definition:

- *A comprehensive list of data sources and collection methods:*
 - Public datasets: Kaggle, World Bank, UNDP, Open Financial Data Platforms.
 - APIs for real-time data: Government databases, regulatory financial APIs, market trends data from financial institutions.
 - Surveys or custom datasets: Collect API feedback from users, survey internal teams on API pain points, and gather system logs for additional insights.

- *Data Cleaning Steps:*
 - Handling missing values using interpolation or data imputation.
 - Standardizing API request timestamps to UTC.
 - Removing duplicate entries based on unique request IDs.
 - Filtering out bot-generated or suspicious API requests.

Data Exploration and Preliminary Analysis:

- *Visualizations of Key Data Trends:*
 - Bar Charts: API request distribution per consumer.
 - Scatter Plots: Relationship between API response time and frequency of requests.
 - Histograms: Distribution of API failures by type and frequency.
 - Box Plots: Identifying anomalies in API latency.
- *Key Trends & Insights Report:*
 - API Usage Patterns: Identifying high-demand endpoints and peak traffic periods.
 - Consumer Behavior Analysis: Categorizing users based on frequency and type of API usage.
 - Performance Insights: Identifying slow APIs and failure trends.
 - Security & Anomaly Detection: Flagging unusual API behaviors and potential security threats.

Data Issue	Resolution
Duplicate API requests	Remove using unique request IDs and timestamps
Outliers in response time	Use box plots to detect and decide if removal is necessary
Non-standardized timestamps	Convert all time data to UTC for consistency
High latency endpoints	Identify and optimize slow-performing APIs

Tools and Technologies:

Component	Tool
Data Ingestion	Python (Log Parsing, API Call Tracking)
Data Processing	Python (Pandas, NumPy, Scikit-learn)
Storage	MySQL (On-Premise)
BI & Visualization	Power BI
Machine Learning	TensorFlow, Scikit-learn
Automation	Python Scripts (CRON Jobs, Task Scheduler)