Workshop No. 3 — Robust System Design and Project Management

Team Members

Luis Mario Ramírez

Santiago Marín

Leonardo Rodríguez

Davidson Sánchez

Project Title:

PartyBarrel Systematica Analysis And Robust Design For The Card image cap.

Refined System Architecture

The IEEE-CIS Fraud Detection Board was readopted according to ISO 9000, CMMI and Six Sigma., by the Redesign, Robust manufacturing model and Scalability. The architecture of the system is modularised meaning minimizing other hardware impact when an upgrade on a single component is required.

Main Components

1. Data Ingestion Module:

- Safely captures, validates raw data (t-Logs. csv, identity. csv).
- It performs schema validation and checksum checks.
- Prevent the data is leak by implementing consistency.

2. Data Preprocessing Module:

- Performs strong imputation of missing data via a CV junction insensitivity.
- Promotes scaling and encoding for reproducibility.
- Comes with version control to monitor pipeline modifications.

3. Feature Engineering Module:

- Create complex temporal and behavioral characteristics (transaction recency, device usage regularity).
- Applies statistical anomaly detection prior to modeling.

4. Modeling and Evaluation Module:

- Utilizes LightGBM with hyperopt and cross validation.
- Evaluation by ROC-AUC ≥ 0.90, Precision@k and Brier Score.
- Includes sensitivity and chaos test to check the stability of the model.

5. Monitoring and Retraining Module:

- Monitors model performance for concept drift.
- Supports automatic retraining with updated data batches.
- Logs deviations with routines for recovering continuity.

6. Design Principles Applied:

- Modularity: It´s all about the points, and these are built to be separate pieces.
- Redundancy: Retries and validation checks are done automatically at every stage.
- Traceability: Model, feature and data transformation versioning.
- Feedback Loops: The error data detected in the feeds moves back to preprocessing and retraining blocks.

Quality and Risk Analysis

Identified Risks and Mitigation Strategies

Risk	Impact	Mitigation Strategy	Monitoring Method
Data Loss or Corruption	Loss of critical training data	Keep all your backups automated and do check some validation within ingestion	Data integrity audits weekly
Model Drift	Performance degradation over time	Incorporate drift monitoring; retraining based on rolling time windows	Real-time monitoring with alert if ROC-AUC < 0.88
Security Breaches	Exposure of user data	Encrypting and anonymizing the data before storage	Penetration testing and access logs
Overfitting / Bias	Reduced generalization	Use temporal validation and regularization	Cross-validation and AUC stability tracking

High Latency in Real-Time Prediction	Delayed responses to fraud	Super charge model deployment with FastAPI and caching	Latency benchmarks during deployment testing

These strategies align with CMMI Level 3 quality assurance and Six Sigma's DMAIC (Define, Measure, Analyze, Improve, Control) methodology to continuously reduce variability and improve process reliability.

Project Management Plan

Team Roles and Responsibilities

Member	Role	Responsibility
Luis Mario Ramírez	Project Manager	Manages workflow and milestones, supervises GitHub repository.
Santiago Marín	Data Engineer	Wrangles ingestion, preprocessing and feature engineering.
Leonardo Rodríguez	ML Specialist	Gets models up and running, does the tuning, sensitivity comparison.
Davidson Sánchez	QA & Deployment Engineer	Watches over metrics, takes control of FastAPI deployment and coordinates risk-response procedures.

Milestones and Deliverables

Phase	Deliverable	Deadline
System Refinement	Updated architecture and risk plan	Week 1
Modeling & Evaluation	ROC-AUC ≥ 0.90 baseline	Week 3
Deployment	FastAPI real-time system	Week 5
Monitoring & Reporting	Drift dashboard + documentation	Week 6

Project Management Tools

- Kanban Board (on GitHub Projects) to keep track of tasks and push around work.
- Gantt Chart: Visualizing milestones and time management.
- Scrum Sprints: Milestone reviews on a weekly basis to review deliverables.
- The source code version control management (github branches of each subsystem, i.e., data, model and API).

Workflow Diagram (Summary)

1. Define Objectives
2. Assign Tasks
3. Develop Modules
4. Integrate & Test
5. Deploy
6. Monitor & Improve

Incremental Improvements

Key lessons from Workshops 1 and 2 that we incorporated to improve robustness:

- Workshop 1: Enhanced awareness and appreciation of systemic interdependencies between datasets and variables. Applied more stringent missing values treatment and validation mechanisms.
- Added chaos and sensitivity analysis into workshop 2 to predict instability and retrain in real-time. Created a training loop to retrain models and added error monitoring dashboards.

New Improvements for Workshop 3:

Some ISO 9000 traceability was introduced: every process and dataset version can be now audited.

Utilized risk-based monitoring with thresholds for data quality and model performance.

Established team management plan with identified milestones and reporting mechanisms.

Reflection

During this phase, we struggled with maintaining model accuracy in real-time. Handling an identity record missing and very imbalanced dataset took several times pre-processing and validation.

With teamwork, module engineering and disciplined project management we delivered a design scalable in size, understandable model-space representation and concurrence with systems engineering standards.

We achieved the planned 0.90 ROC-AUC value in the test, indicating that the design was robust and of high quality.

References

IEEE-CIS Fraud Detection, Kaggle (2025).

Saltelli, A. et al. Sensitivity Analysis in Practice. Wiley, 2004.

Gleick, J. Chaos: Making a New Science. Penguin Books, 1987.

ISO 9001:2015 Quality Management Systems – requirements.

CMMI Institute, Capability Maturity Model Integration. 2023.