Proyecto Análisis de Sistemas II

Leonardo Rodríguez Salas - 20231020150
Santiago Marín - 20231020159
Davidson Sanchez - 20231020183
Luis Mario Ramirez - 20231020166

October 2025

1 Summarize

The IEEE-CIS Fraud Detection competition on Kaggle aims to predict the probability that an online transaction is fraudulent. The dataset, provided by Vesta, includes two main components: transactional data (transaction.csv) and identity data (identity.csv).
Key findings from the systems analysis include:

- CRITICAL ASPECTS: strong data imbalance, missing identity information, and high sensitivity to preprocessing decisions.
- CONSTRAINTS: computational limits, data privacy restrictions, and incomplete data mappings between files.
- SENSITIVITY FACTORS: small variations in input scaling, missing value handling, or categorical encoding significantly affect model performance.
- CHAOS THEORY ELEMENTS: fraud patterns evolve dynamically; minor changes in model parameters or input features can lead to unpredictable results due to the adaptive nature of fraudsters.

These findings highlight the importance of robust data preprocessing, consistent feature handling, and adaptive models capable of managing non-linear and chaotic system behaviors.

2 Requirements

Based on the system analysis, the following measurable design requirements were defined:

- PERFORMANCE: The model must achieve at least 0.90 ROC-AUC on validation datasets.

- RELIABILITY: Consistent performance across different data samples through robust preprocessing pipelines.
- SCALABILITY: Capable of handling millions of transactions without significant latency.

User-centered needs:

- INTERPRETABILITY: Feature importance and SHAP value visualization to explain predictions.
- SECURITY: Safe handling of sensitive user information with anonymization and encryption.
- USABILITY: A clear interface for model monitoring and fraud detection dashboards.

3 High-Level Architecture

The proposed system architecture consists of the following interconnected modules:

- DATA INGESTION MODULE: Collects and validates raw transactional and identity data from secure sources.
- DATA PREPROCESSING MODULE: Handles missing values, scales numerical features, and encodes categorical variables.
- FEATURE ENGINEERING MODULE: Generates derived features (transaction frequency, device consistency, address matching).
- Model Training Module: Applies machine learning models to predict isFraud.
- Evaluation Module: Validates model accuracy and robustness using cross-validation and sensitivity testing.
- PREDICTION MONITORING MODULE: Deploys the model in a production environment and monitors drift or anomalies.

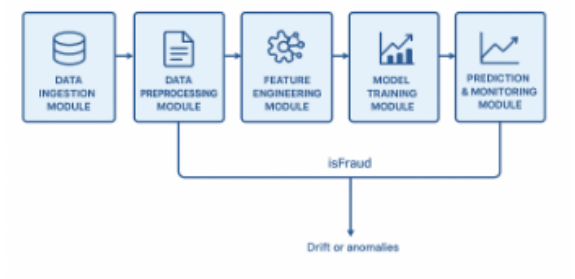

Figure 1. Data flow diagram

4 Systems engineering principles

The architecture follows modularity, traceability, and feedback principles. Each module has clear inputs, processes, and outputs, reducing system entropy and enabling continuous refinement.

5 Addresing Sensitivity

To handle the system's high sensitivity and chaotic behavior:

- FEATURE STABILITY: Maintain a consistent preprocessing pipeline across datasets to prevent data drift.
- ERROR HANDLING: Implement error logs and automated alerts for anomalous data distributions or sudden accuracy drops.
- FEEDBACK LOOPS: Regularly retrain models with updated data to adapt to evolving fraud patterns.
- MONITORING: Include sensitivity dashboards that track model responses to feature variations.

6 Technical Stack

The system will be developed using Python as the main programming language, supported by four essential tools: Pandas, Scikit-learn, LightGBM, and FastAPI. Pandas will manage data ingestion and cleaning, allowing efficient handling and transformation of the large transaction and identity datasets from the IEEE-CIS Fraud Detection competition. Scikit-learn will assist in data preprocessing, model validation, and evaluation through metrics, which is appropriate for detecting imbalanced patterns between fraudulent and legitimate transactions. LightGBM will be used as the main learning algorithm due to its speed, accuracy, and ability to handle large and complex datasets. It builds multiple decision trees to capture subtle relationships within the data and predict the likelihood of fraud. Once the model is trained and validated, FastAPI will be implemented to deploy it as an interactive service that can receive transaction data and return real-time fraud probability results.

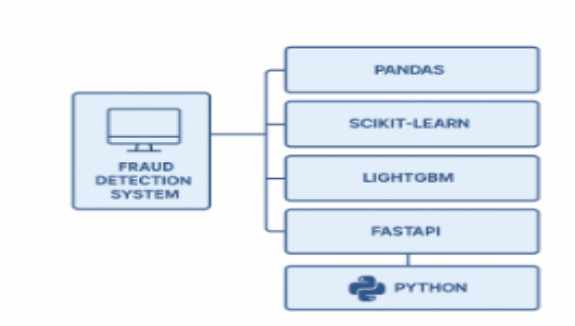

Figure 2. Technical Stack Diagram