Banking Transaction Fraud Detection System

Submitted By:

Name: Jyotiranjan Mahapatra

BANK FRAUD DETECTION SYSTEM

Introduction

Banking systems process millions of transactions daily. With the rapid growth of digital payments, the risk of fraudulent transactions has increased significantly. Fraud detection systems help financial institutions identify suspicious activities and prevent financial losses.

This project focuses on building a Bank Fraud Detection System using Python, Apache Spark, Snowflake, and Streamlit to detect high-risk transactions and visualize fraud insights.

Problem Statement

Financial fraud in banking transactions leads to major economic losses. Manual monitoring of transactions is inefficient and time-consuming.

Therefore, there is a need for an automated system that:

- Processes large transaction datasets
- Detects suspicious transactions
- Provides real-time fraud insights
- Visualizes fraud analytics interactively

Objectives

- To build an ETL pipeline for transaction data.
- To perform fraud detection using rule-based logic.
- To process large datasets using Apache Spark.
- To visualize fraud statistics using Streamlit dashboard.
- To demonstrate real-world data engineering workflow.

System Architecture

Architecture Flow:

Raw Data → ETL → Snowflake → Spark Processing → Dashboard

Technologies Used

Technology	Purpose
Python	Core programming
Pandas	Data processing
Apache Spark	Batch & Streaming processing
Snowflake	Data warehousing
Streamlit	Dashboard visualization
Virtual Environment	Dependency management

Dataset Description

The system uses banking transaction data containing:

- Transaction ID
- Customer ID
- Merchant ID
- Transaction Time
- Transaction Amount
- Payment Channel

Example:

transaction_id	amount	channel
TX101	5000	UPI
TX102	150000	CARD

Methodology

Step 1: Data Ingestion

Raw transaction data is stored in CSV format.

Step 2: ETL Process

Data is extracted, transformed, and optionally loaded into Snowflake.

Step 3: Fraud Detection Logic

Transactions are marked as fraud if:

amount > 200000

This rule identifies unusually high-value transactions.

Step 4: Spark Processing

- Batch processing for historical fraud analysis
- Streaming processing for real-time fraud detection

Step 5: Dashboard Visualization

Streamlit dashboard displays:

- Total transactions
- Fraud transactions
- Fraud percentage
- Fraud by channel

Results

The system successfully:

- Identifies high-value suspicious transactions
- Calculates fraud percentage
- Visualizes fraud distribution by payment channel
- Displays transaction-level details

Advantages

- Simple and scalable architecture
- Real-time processing capability
- Easy visualization using Streamlit
- Can be extended with Machine Learning

Limitations

- Uses rule-based detection (not ML-based)
- Fraud logic based only on transaction amount
- No user authentication system

Future Enhancements

- Implement Machine Learning fraud model
- Add anomaly detection algorithms
- Integrate Kafka for real-time streaming
- Deploy on cloud (AWS / Azure)
- Add role-based access control

Conclusion

The Bank Fraud Detection System demonstrates a complete data engineering workflow, including ETL processing, distributed computation using Spark, and dashboard visualization using Streamlit.

This project can be further enhanced with machine learning techniques to improve fraud detection accuracy.