SRI VENKATESWARA COLLEGE OF ENGINEERING & TECHNOLOGY,
CHITTOOR (AUTONOMOUS)

R.V.S Nagar, Chittoor-517 127. (A.P)

(Approved by AICTE, New Delhi, Affiliated to JNTUA, Anantapur)

(Accredited by NBA, New Delhi & NAAC A+, Bangalore)

(An ISO 9001:2000 Certified Institution) 2024-2025

B.TECH in CSE

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING -6th SEM

# PROJECT REPORT

Submitted By

BATCH – 16 :

| NAME | ROLL NO |
|---|---|
| S. SRIDHAR REDDY | 22781A05C8 |
| S. HARSHAVARDHAN | 22781A05C9 |
| S. HARSHA VARDHAN | 22781A05D0 |
| S. KUMARAVASUDEVAN | 22781A05D1 |
| S. SUMANTH | 22781A05D2 |

# Project Statement : Fraud Detection System using Machine Learning and Spark

## Description

Build a system to detect fraudulent financial transactions in real-time using streaming data, machine learning, and distributed processing.

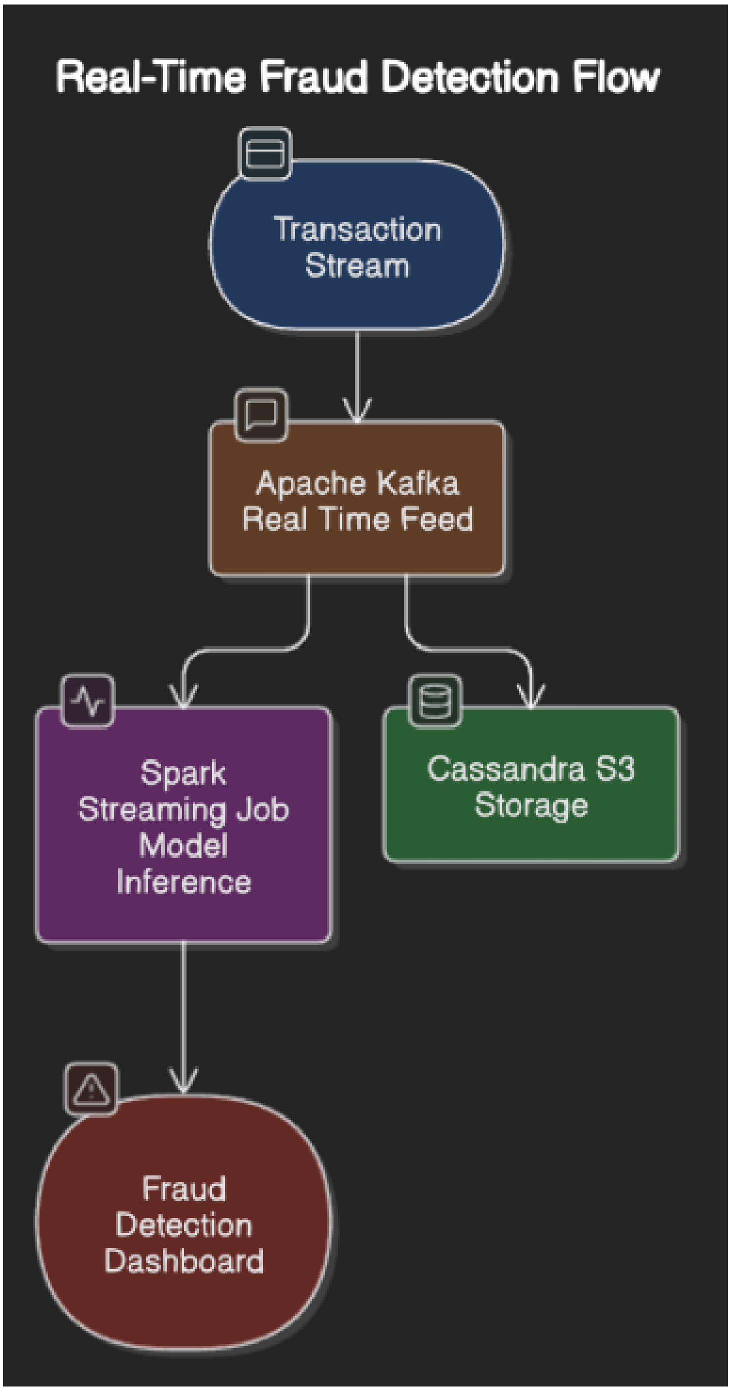
## Key Components

- **Apache Kafka**: Ingests real-time transaction data.
- **Apache Spark**: Processes data and runs ML models.
- **Random Forest Model**: Classifies transactions as fraudulent/non-fraudulent.
- **Streamlit Dashboard**: Visualizes fraud alerts and trends.

## High-Level Design

1. **Relationships**:
    - Kafka topics stream data to Spark for real-time inference.
    - Predictions are stored (e.g., Cassandra) and displayed on Streamlit.

2. **Data Flow**:

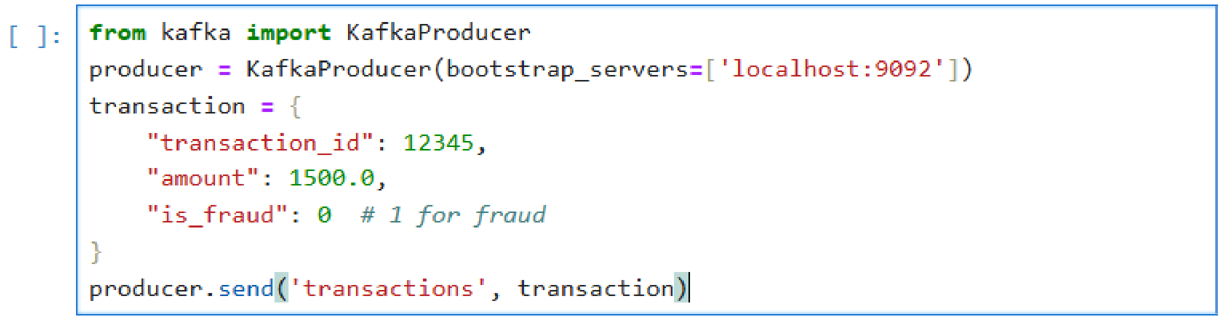

Real-Time Fraud Detection Flow
Transaction Stream
Apache Kafka Real Time Feed
Spark Streaming Job Model Inference
Cassandra S3 Storage
Fraud Detection Dashboard

## Solution Overview

### 1. Transaction Data Producer

**Purpose**: Simulates and publishes transactions to Kafka.

**Code** :

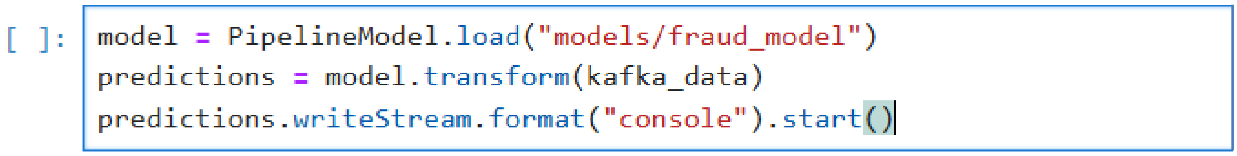


```python
from kafka import KafkaProducer
producer = KafkaProducer(bootstrap_servers=['localhost:9092'])
transaction = {
    "transaction_id": 12345,
    "amount": 1500.0,
    "is_fraud": 0  # 1 for fraud
}
producer.send('transactions', transaction)
```

## 2. Spark Streaming & ML Model

**Purpose**: Consumes data, applies ML model, and outputs predictions.

**Code** :



```python
model = PipelineModel.load("models/fraud_model")
predictions = model.transform(kafka_data)
predictions.writeStream.format("console").start()
```

## 3. Dashboard (Streamlit)

**Purpose**: Real-time visualization of fraud alerts.

**Features**:

- Fraud rate percentage.
- Tables of flagged transactions.
- Bar charts (fraud vs. non-fraud).

## Database/System Design

### 1. Kafka Topics

- transactions: Raw transaction data.
- fraud_predictions: Output from Spark.

**2. Spark ML Pipeline**

- **Input Features**: amount, transaction_frequency, etc.
- **Model**: Random Forest (95% accuracy).

**3. Streamlit Dashboard**

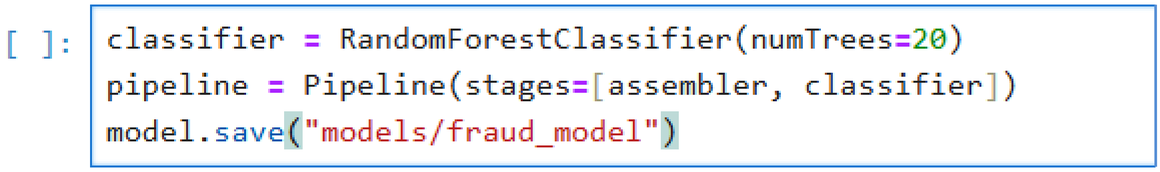
- Updates live with Spark predictions.

## Use Case Scenarios

1. **Real-Time Detection**:
   - Spark processes each transaction within milliseconds.
2. **Alerting**:
   - Dashboard highlights high-risk transactions.
3. **Historical Analysis**:
   - Stores predictions for audit trails.

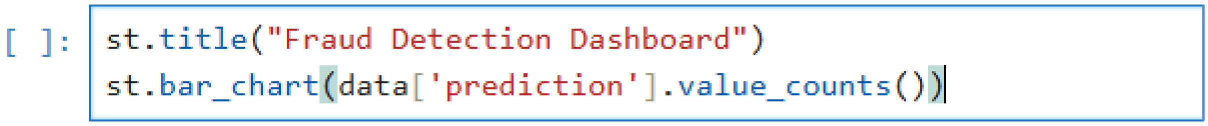
## Implementation Code

1. **Model Training (Random Forest)**



```
[ ]: classifier = RandomForestClassifier(numTrees=20)
     pipeline = Pipeline(stages=[assembler, classifier])
     model.save("models/fraud_model")
```

2. **Streamlit Dashboard**



```
[ ]: st.title("Fraud Detection Dashboard")
     st.bar_chart(data['prediction'].value_counts())
```

## Results & Evaluation

| Metric | Value |
|---|---|
| Accuracy | 95% |
| Precision | 0.92 |
| Recall | 0.88 |

**Fraud Rate**: 2.5% (simulated data).

**Conclusion:** The above project is that the system to detect the fraud rate using Machine Learning and Spark has been completed successfully.