National College of Ireland

Configuration Manual

MSc Research Project
Master's In Data Analytics

Jennifer Ullasa Kumar
Student ID: x23286091

School of Computing
National College of Ireland

Supervisor: Eric Gyamfi

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Jennifer Ullasa Kumar

Student ID: x23286091

Programme: Master In Data Analytics **Year:** 2025

Module: Research Practicum

Lecturer: Eric Gyamfi

Submission Due Date: 15/09/2025

Project Title: Hybrid Real Fraud Detection

Word Count: 764 **Page Count:** 7

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Jennifer Ullasa Kumar

Date: 11/08/2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Jennifer Ullasa Kumar
x23286091

1 System Configuration

- OS: macOS (works on Linux/Windows too)
- RAM: 8–16 GB
- Python: 3.10+ in a virtual env
- Docker: Desktop installed (for Kafka/Zookeeper via docker compose)
- Kafka: brought up by docker-compose.yml
- Required Python packages

  pip install kafka-python streamlit pandas numpy scikit-learn joblib river altair

- Project files (root folder):
  - docker-compose.yml – Kafka + Zookeeper
  - start_all.sh – helper to launch everything (Mac: opens new Terminal tabs)
  - kafka_producer.py – streams creditcard.csv into Kafka
  - kafka_consumer.py – scores messages, writes logs, sends alerts
  - app.py – Streamlit dashboard
  - Models: rf_model.joblib, scaler.joblib
  - Data: creditcard.csv
  - Outputs (created at runtime):

    stream_results.csv, kafka_alerts.csv, metrics_window.csv, kafka_drift_log.csv

2 Execution Steps

One Shot Laucher

```
chmod +x start_all.sh
./start_all.sh
```

Manual:

Terminal 1:

```
docker compose up -d && docker compose ps
```

Termial 2: Consumer

```
source .venv/bin/activate
python3 kafka_consumer.py
```

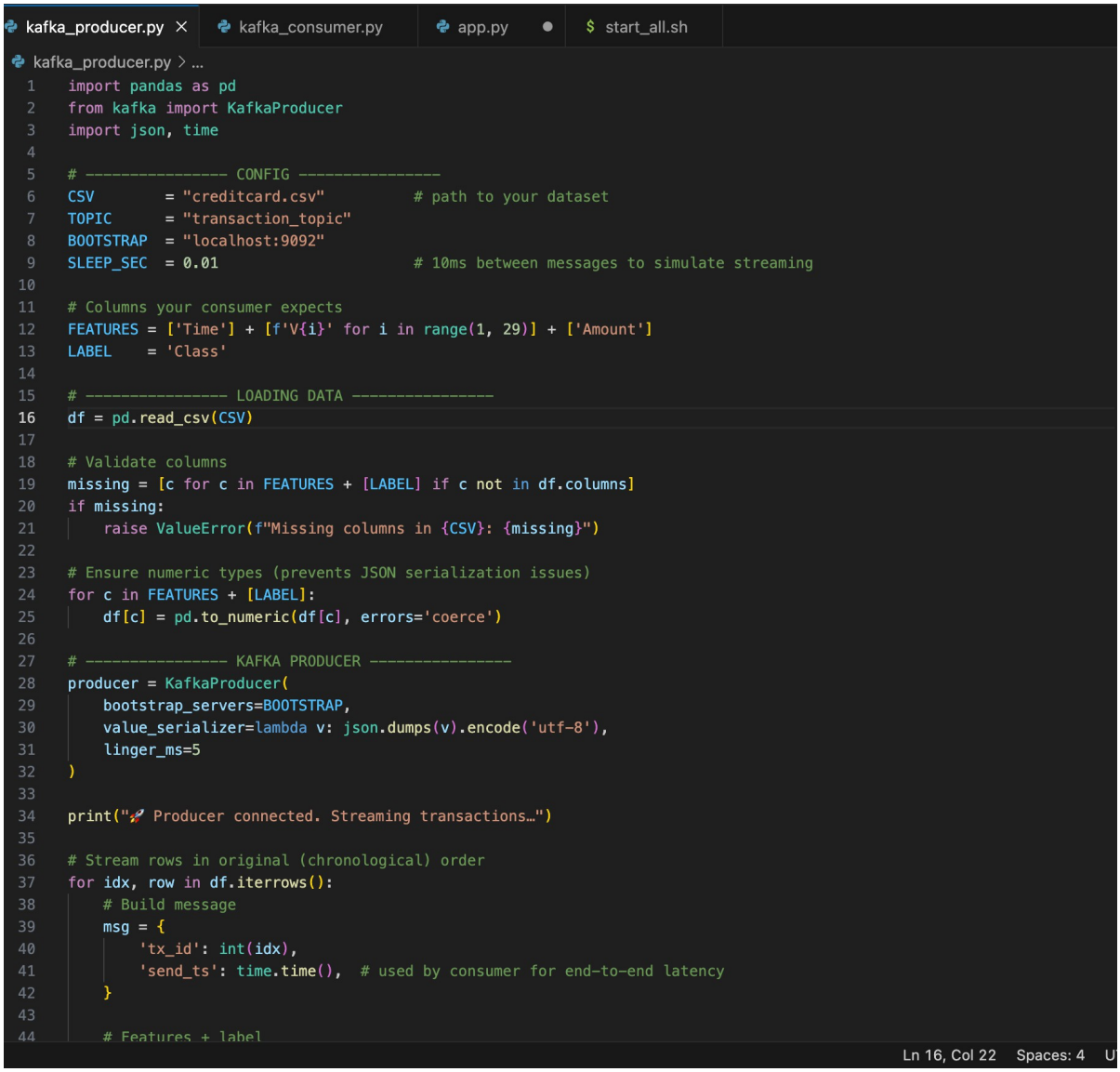
Terminal 3: Producer

```
source .venv/bin/activate
python3 kafka_producer.py
```

Terminal 4: Dashboard
source .venv/bin/activate
streamlit run app.py

3 Section 3

3.1 kafka_producer.py

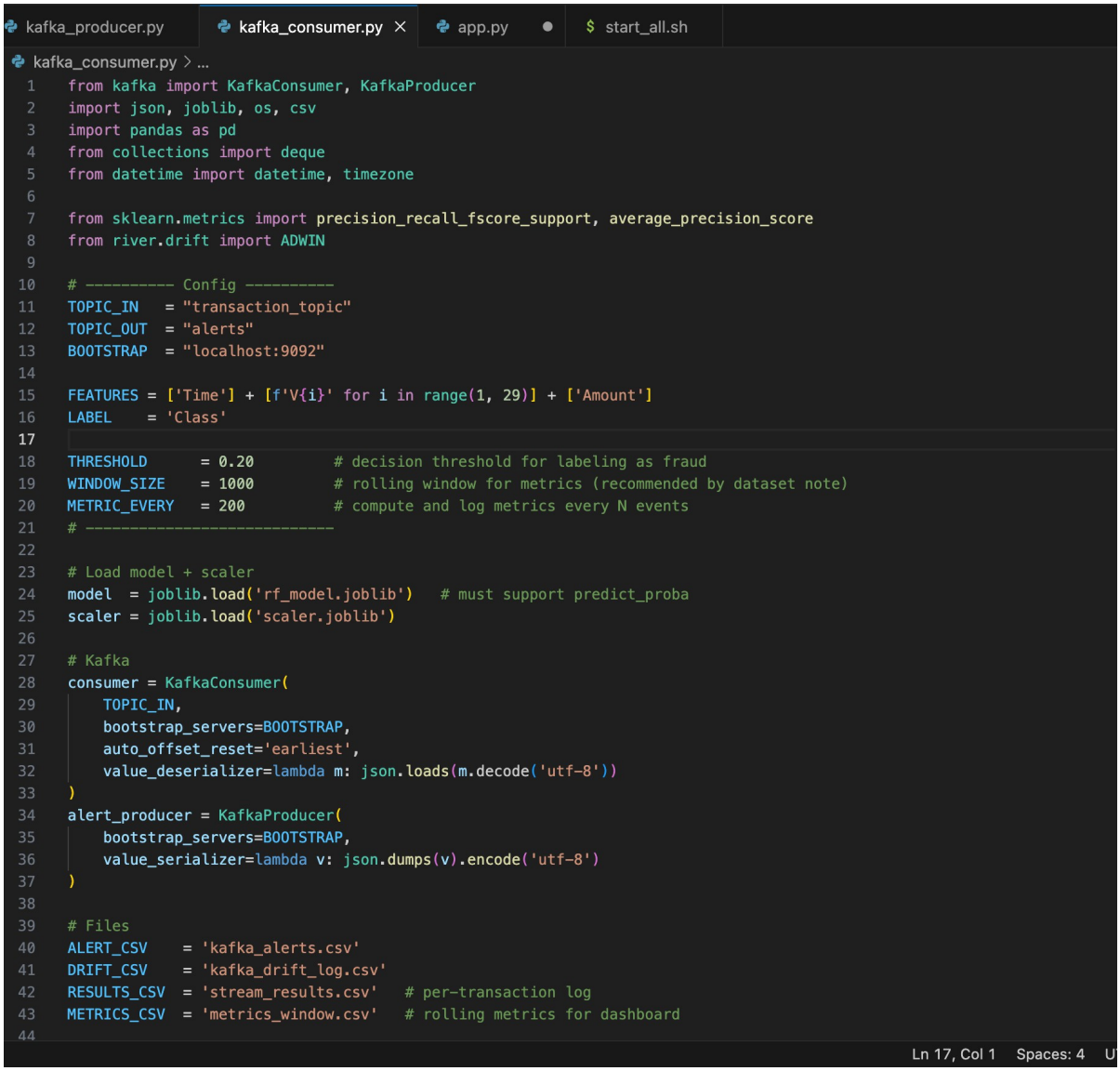


Purpose: read creditcard.csv and publish each transaction to Kafka.

Core flow:

- Config: TOPIC="transaction_topic", BOOTSTRAP="localhost:9092".
- Validates columns ['Time', V1…V28, 'Amount', 'Class'].
- Serializes each row as JSON: includes tx_id, timestamps, features, and label.
- Sends to Kafka with a tiny sleep (0.01s) to simulate real-time.
- Why it matters: gives a controlled, replayable stream to test the online pipeline.

3.2 kafka_consumer.py



Purpose: consume messages, score fraud probability, emit alerts, log metrics.

Core flow:

- Config: input topic transaction_topic, output topic alerts, THRESHOLD=0.20.
- Loads rf_model.joblib and scaler.joblib, uses predict_proba on features.
- Writes every prediction to stream_results.csv with columns:

  tx_id, pred, label, p_fraud, ts.

- If p_fraud >= THRESHOLD, sends an alert to Kafka and appends to kafka_alerts.csv.
- Maintains a rolling window (WINDOW_SIZE=1000) and every METRIC_EVERY=200 events logs precision, recall, F1, and average precision to metrics_window.csv.
- Runs ADWIN drift detection on errors; on change, logs a timestamp to kafka_drift_log.csv.

- Why it matters: closes the real-time loop with decisions, observability, and drift flags.

3.3 C. Dashboard app.py

kafka_producer.py | kafka_consumer.py | app.py | start_all.sh

app.py > ...

```python
import os
import time
import pandas as pd
import numpy as np
import streamlit as st
import altair as alt

# Prefer average_precision_score (AUPRC) but fall back gracefully
try:
    from sklearn.metrics import average_precision_score, precision_recall_curve, auc, precision_score, recall_score
    HAVE_SK = True
except Exception:
    HAVE_SK = False

# ---------- Config ----------
RESULTS_CSV = "stream_results.csv"   # tx_id,pred,label,p_fraud,ts
ALERTS_CSV  = "kafka_alerts.csv"     # tx_id,score,amount,time
DRIFT_CSV   = "kafka_drift_log.csv"  # DriftAt

st.set_page_config(
    page_title="Real-Time Fraud Detection — Live Ops",
    layout="wide",
    page_icon="🚨",
)

# ---------- Helpers ----------
@st.cache_data(show_spinner=False)
def safe_read_csv(path: str, nrows: int | None = None) -> pd.DataFrame:
    """Read a CSV that is being appended to, skipping malformed lines and
    forcing the exact 5 columns we expect for stream_results.csv.
    """
    if not os.path.exists(path) or os.path.getsize(path) == 0:
        return pd.DataFrame(columns=["tx_id", "pred", "label", "p_fraud", "ts"])

    try:
        df = pd.read_csv(
            path,
            header=None,
            names=["tx_id", "pred", "label", "p_fraud", "ts"],
            on_bad_lines="skip",
            engine="python",          # more forgiving for trailing writes
            nrows=nrows,
        )
    except Exception:
```

Ln 266, Col 15 Spaces: 4 U

Purpose: live operations view in Streamlit.

Core flow:

- Page config: wide layout, title “Real-Time Fraud Detection – Live Ops”.
- Safely reads append-only CSVs with a cached loader that skips malformed lines.
- Panels:
    - Recent Alerts table (kafka_alerts.csv) – last few high-risk transactions.
    - Performance chart (metrics_window.csv) – precision, recall, F1 over time (Altair line chart).
    - Latest transactions table (stream_results.csv) – tail view with p_fraud.
- Why it matters: gives you immediate feedback on model behavior and alert volume without digging into logs.

References

1. adimov, E. and Birihanu, E. (2025) ‘Real-time suspicious detection framework for financial data streams’, *International Journal of Information Technology*.
2. Alang, K. S. and Kushwaha, A. S. (2025) ‘Stream processing with Apache Kafka: Real-time data pipelines’, *International Journal of Research in Modern Engineering & Emerging Technology*.
3. Bifet, A. and Gavaldà, R. (2007) ‘Learning from time-changing data with adaptive windowing’, in *Proceedings of the 2007 SIAM International Conference on Data Mining*.
4. Bian, S., Li, C., Fu, Y., Ren, Y., Wu, T., Li, G.-P. and Li, B. (2021) ‘Machine learning-based real-time monitoring system for smart connected worker to improve energy efficiency’, *Journal of Manufacturing Systems*.