

SKaMP. Tests

The goal of this report is to provide information on the performed testing of data acquisition, data pre-processing, batch processing and streaming processing.

GitHub repository: <https://github.com/salveendutt/Big-Data-Analytics>.

1 Test Scenarios

Test objective	Steps	Expected Result	Actual Result
Verify data incoming from stream API	1. Start the server using start_containers.bat; 2. Navigate to http://localhost:5000	Incoming data is available on /data/0	Passed. The screenshot is provided in Figure 3
Verify correct setup of the stream and data preprocessing functions	Run 'pytest' from the root folder	Data stream is configured as expected; Incoming data is not null; Returned status code - 200. Preprocessing utils return transformed data as expected	Passed. The screenshot is provided in Figure 12
Verify the correct setup of Nifi - HDFS/Kafka flow	Run the containers - follow steps in README.md	Data flows from streamin API to Kafka topics and Hive tables	Passed. The screenshot is provided in Figure 4, 5, 6, 7
Verify the correct setup of batch processing	Run the containers - follow steps in README.md	Views are available in the Cassandra tables	Passed. The screenshot is provided in Figure 8, 9, 10
Verify the correct setup of streaming processing	Run the containers - follow steps in README.md	Views are available in the Cassandra tables	Passed. The screenshot is provided in Figure 11
Verify that Superset correctly connects to Cassandra through TrinoDB and displays data in dashboards	Run the necessary containers (start_containers) in 'scripts' folder and observe the charts in Superset	Data is visible in Superset charts	PASSED. Screenshots are visible in Figure x and x

Table 1: Test scenarios

Test objective	Steps	Expected Result	Actual Result
Verify correct data processing of dataset 1	Run 'pytest' from the root folder	Feature 'type' is correctly transformed into numeric value (5 cases); Feature 'is-Merchant' is correctly prepared (2 cases)	PASSED. The screenshot is provided in Figure 12
Verify correct data processing of dataset 2	Run 'pytest' from the root folder	Numeric boolean values are transformed to int from float (4 cases)	PASSED. The screenshot is provided in Figure 12
Verify correct data processing of dataset 3	Run 'pytest' from the root folder	Feature 'entry_mode' is correctly transformed into numeric value (4 cases); Unnecessary features are omitted.	PASSED. The screenshot is provided in Figure 12
Verify correct data processing of dataset 4	Run 'pytest' from the root folder	Features 'Amount', 'Class' are renamed to 'amount' and 'is-Fraud'; Extra features are removed	PASSED. The screenshot is provided in Figure 12

Table 2: Data pre-processing tests

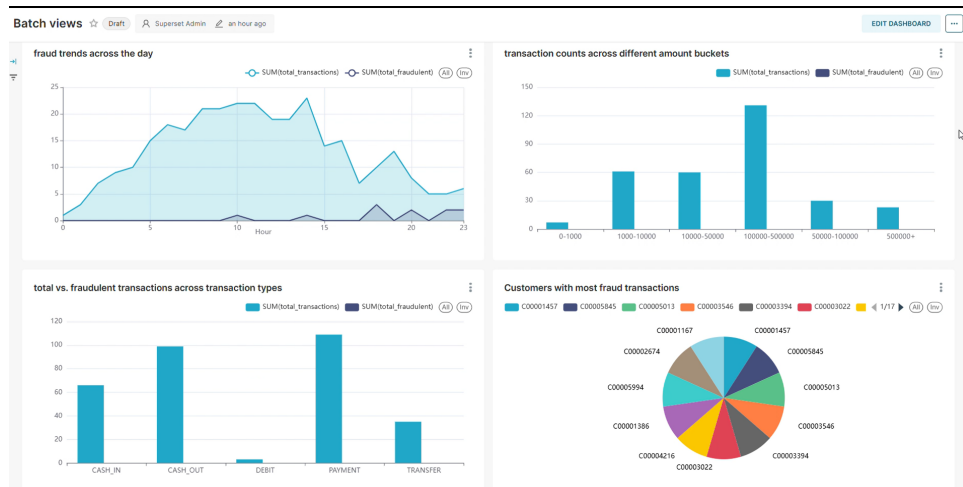


Figure 1: Batch Dashboard



Figure 2: Speed Dashboard

```
localhost:5000/data/0
{
  "amount": "312003.01",
  "isFlaggedFraud": "0",
  "isFraud": "0",
  "nameDest": "C1845208133",
  "nameOrig": "C1852599404",
  "newbalanceDest": "1545311.79",
  "newbalanceOrig": "8663310.08",
  "oldbalanceDest": "1857314.8",
  "oldbalanceOrg": "8351307.07",
  "step": "44",
  "type": "CASH_IN"
}
```

Figure 3: Data incoming via the stream

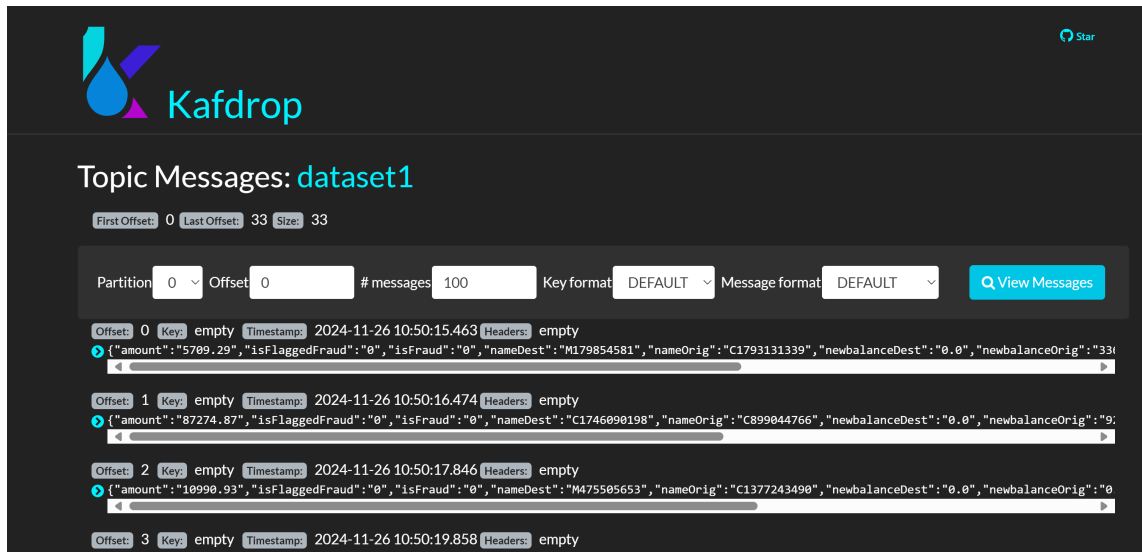


Figure 4: Kafka Dataset1

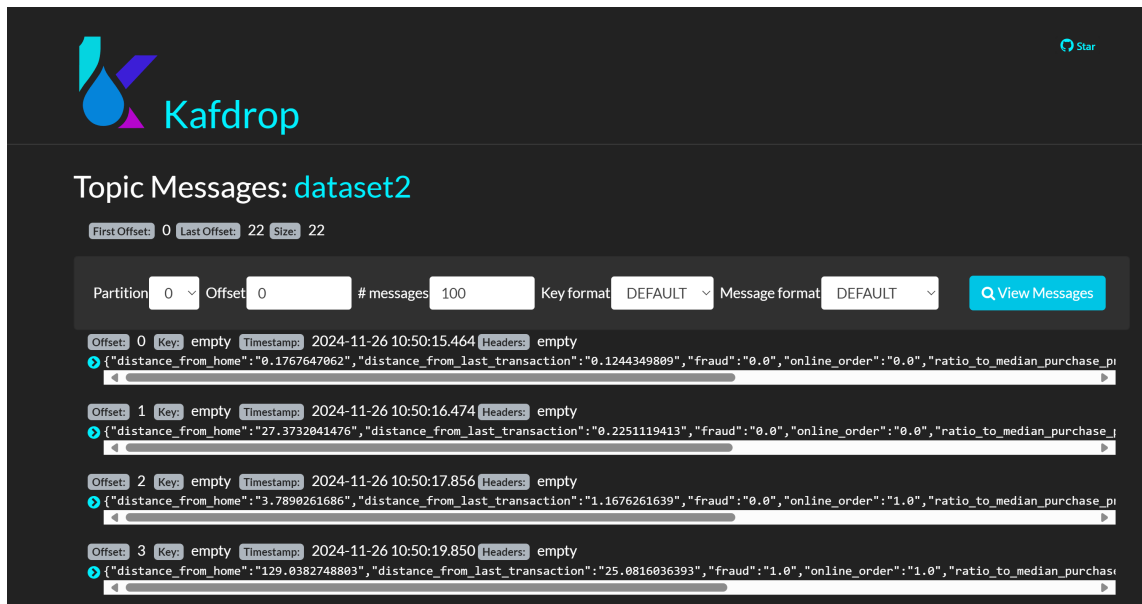


Figure 5: Kafka Dataset2

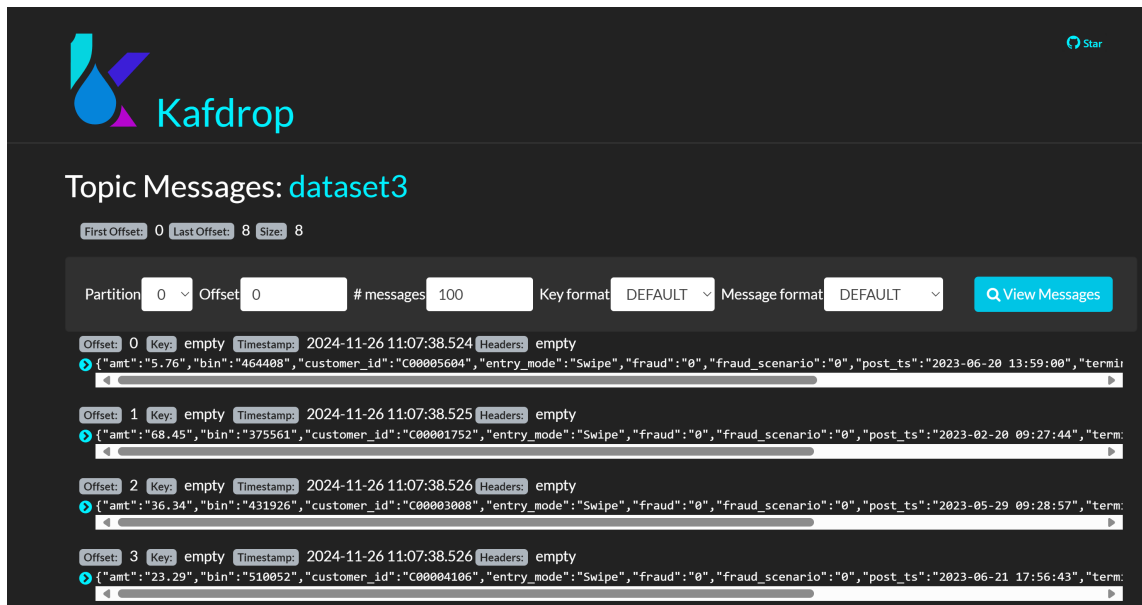


Figure 6: Kafka Dataset3

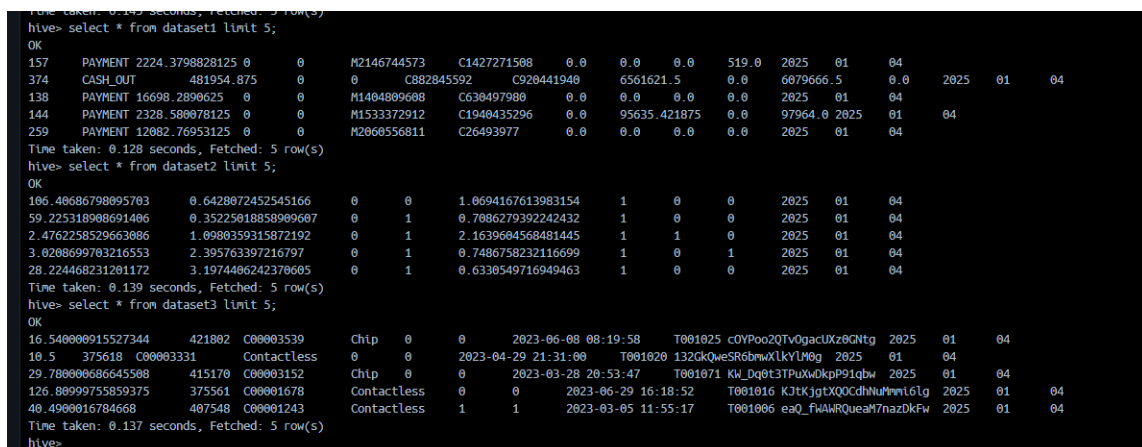


Figure 7: Hive data

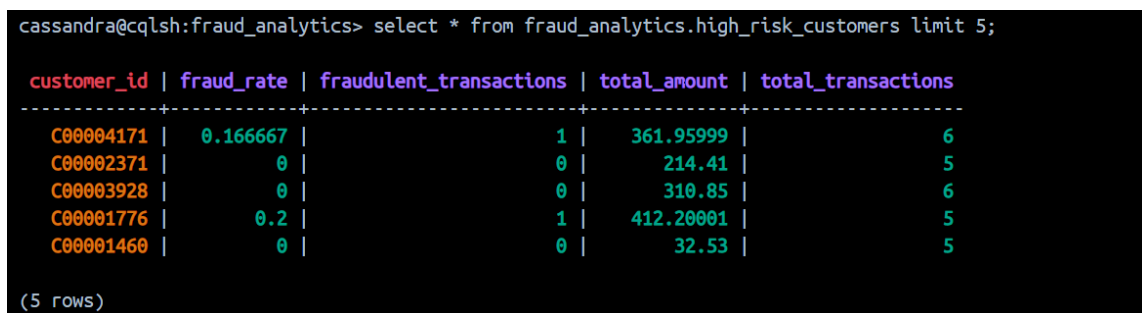


Figure 8: Cassandra batch processing views

```
cassandra@cqlsh:fraud_analytics> select * from fraud_analytics.hourly_fraud_stats limit 5;
```

hour	avg_amount	total_fraudulent	total_transactions
23	59.18323	6	31
5	54.17043	1	117
10	55.43596	4	230
16	55.149	6	190
13	56.83678	3	242

(5 rows)

Figure 9: Cassandra batch processing views

```
cassandra@cqlsh:fraud_analytics> select * from fraud_analytics.fraud_by_transaction_type limit 5;
```

type	avg_amount	fraud_rate	total_fraudulent	total_transactions
TRANSFER	9.3788e+05	0.007463	2	268
CASH_OUT	1.7427e+05	0.001847	2	1083
DEBIT	5886.82766	0	0	25
CASH_IN	1.7194e+05	0	0	708
PAYMENT	12990.00553	0	0	1165

(5 rows)

Figure 10: Cassandra batch processing views

```
(21 rows)
cassandra@cqlsh> select * from fraud_analytics.real_time_predictions ;
```

transaction_id	amount	customer_id	ensemble_fraud_probability	is_fraud	model1_fraud_probability	model2_fraud_probability	model3_fraud_probability	model_version	prediction_timestamp	tran
C1596528350_11	91877.21804	C0001681	0.6	True	1	0.6	0.2	20250104_173150	2025-01-04 17:35:21.994000+0000	
C096684087_41	3558.73999	C0005700	0.333333	False	0.2	0.6	0.2	20250104_173150	2025-01-04 17:35:44.537000+0000	
C1587839118_279	18156.94022	C0005477	0.566667	True	1	0.5	0.2	20250104_173150	2025-01-04 17:35:00.415000+0000	
C778154902_370	16014.5	C0005723	0.466667	False	1	0.2	0.2	20250104_173150	2025-01-04 17:33:01.677000+0000	
C1433374494_257	75861.46004	C00083221	0.566667	True	1	0.5	0.2	20250104_173150	2025-01-04 17:34:42.974000+0000	
C367402113_275	2.159e+05	C00083272	0.6	True	1	0.6	0.2	20250104_173150	2025-01-04 17:34:55.236000+0000	
C948213176_278	1.3629e+05	C00082423	0.6	True	1	0.6	0.2	20250104_173150	2025-01-04 17:34:40.271000+0000	
C1702298778_355	11935.81953	C00083800	0.6	True	1	0.6	0.2	20250104_173150	2025-01-04 17:33:24.820000+0000	
C1070163376_254	1.7259e+05	C00081170	0.5	False	1	0.3	0.2	20250104_173150	2025-01-04 17:34:58.507000+0000	

Figure 11: Cassandra stream processing views

```
===== test session starts =====
platform win32 -- Python 3.13.0, pytest-8.3.3, pluggy-1.5.0 -- C:\ProgramFiles\Anaconda3\envs\bigdata13\python.exe
cachedir: .pytest_cache
rootdir: C:\home\WUT\Semester_3\BigData\Big-Data-Analytics
collected 12 items

services/streaming_simulation/test_streaming_simulation.py::StreamingSimulationTestCase::test_data_stream PASSED [ 8%]
tests/data_utils/test_utils.py::test_preprocess_1_payment PASSED [ 16%]
tests/data_utils/test_utils.py::test_preprocess_1_cash_in PASSED [ 25%]
tests/data_utils/test_utils.py::test_preprocess_1_cash_out PASSED [ 33%]
tests/data_utils/test_utils.py::test_preprocess_1_debit PASSED [ 41%]
tests/data_utils/test_utils.py::test_preprocess_1_unknown PASSED [ 50%]
tests/data_utils/test_utils.py::test_preprocess_row_2 PASSED [ 58%]
tests/data_utils/test_utils.py::test_preprocess_3_contactless PASSED [ 66%]
tests/data_utils/test_utils.py::test_preprocess_3_chip PASSED [ 75%]
tests/data_utils/test_utils.py::test_preprocess_3_swipe PASSED [ 83%]
tests/data_utils/test_utils.py::test_preprocess_3_unknown PASSED [ 91%]
tests/data_utils/test_utils.py::test_preprocess_row_4 PASSED [100%]

===== 12 passed in 0.57s =====
```

Figure 12: Unit testing result

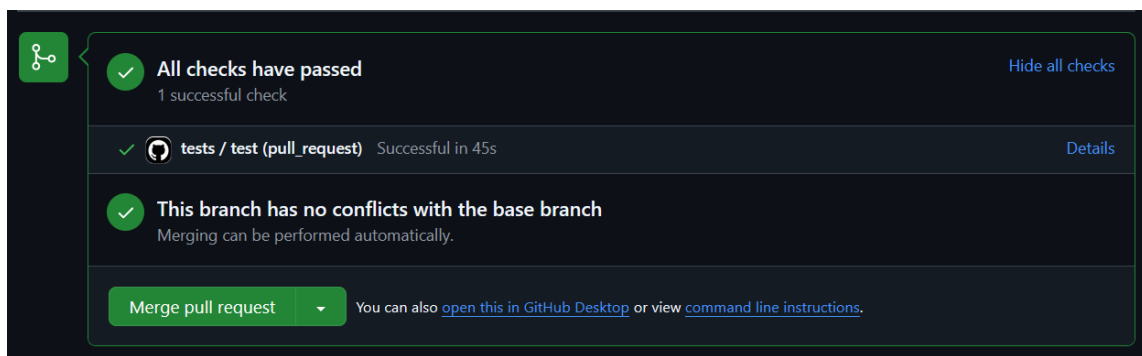


Figure 13: GitHub checks before merge