SKaMP. Tests

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

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

1 Data acquisition

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 Fig.1 and Fig.2
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 Fig. 8
Verify correct setup of NiFi, Kafka and Cassandra	Run the containers - follow steps in README.md	Data flows from streamin API to Kafka topics and Cassandra tables	Passed. The screenshot is provided in Fig. 2, 3, 4, 5, 6, 7

Table 1: Data acquisition tests

2 Data pre-processing

Unit testing is included in the CI/CD pipeline on GitHub and must be successful before any merge into the main branch.

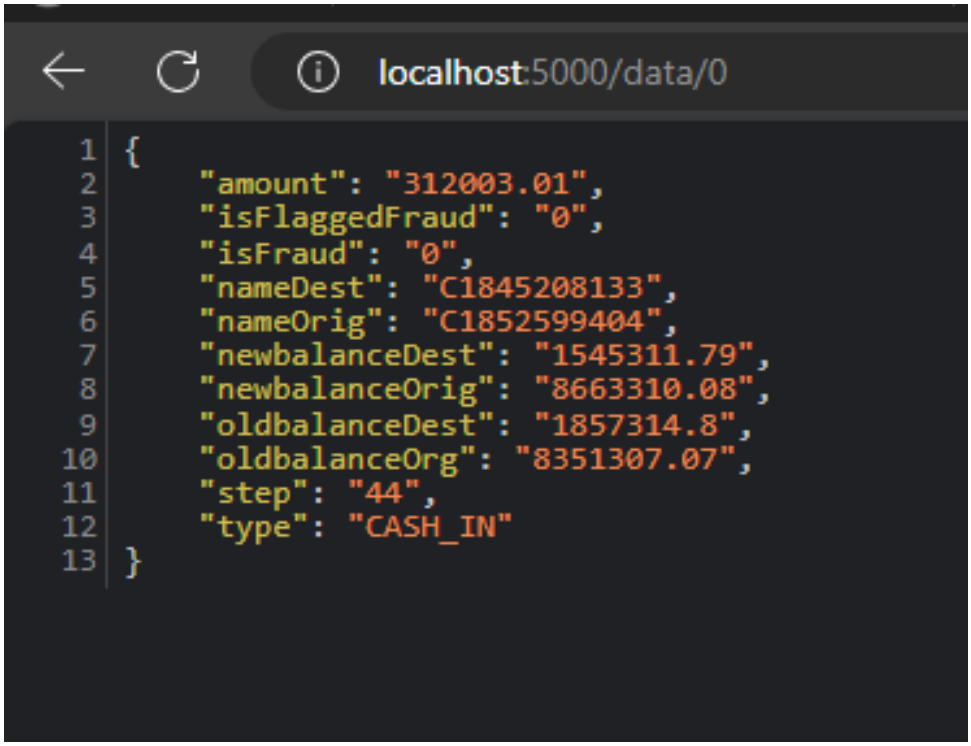


Figure 1: Data incoming via the stream

Figure 2: Kafka Dataset1

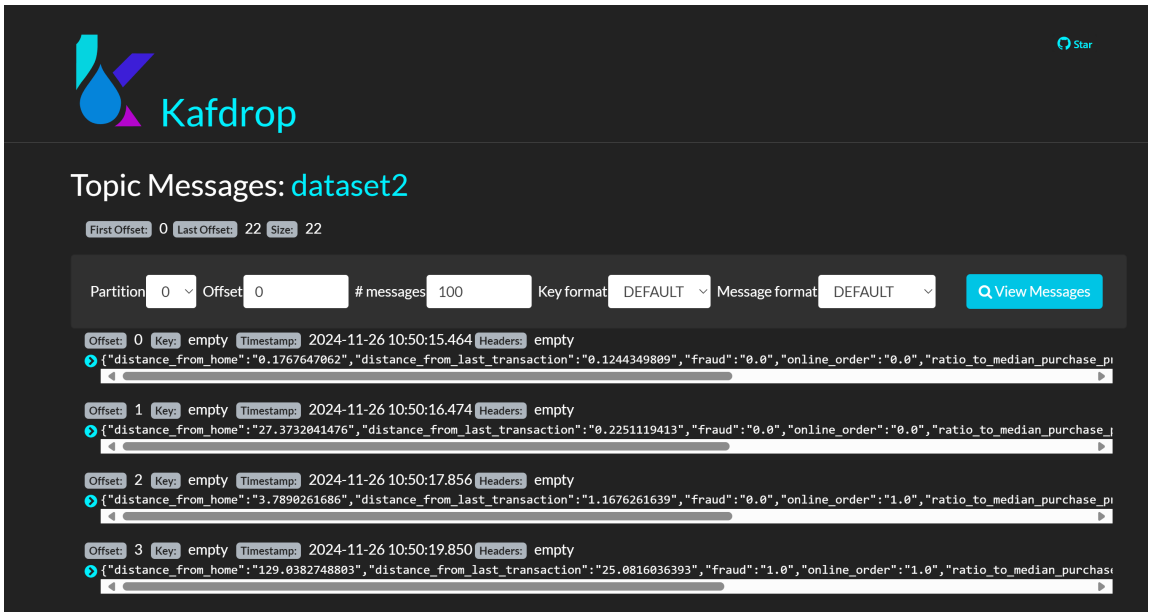


Figure 3: Kafka Dataset2

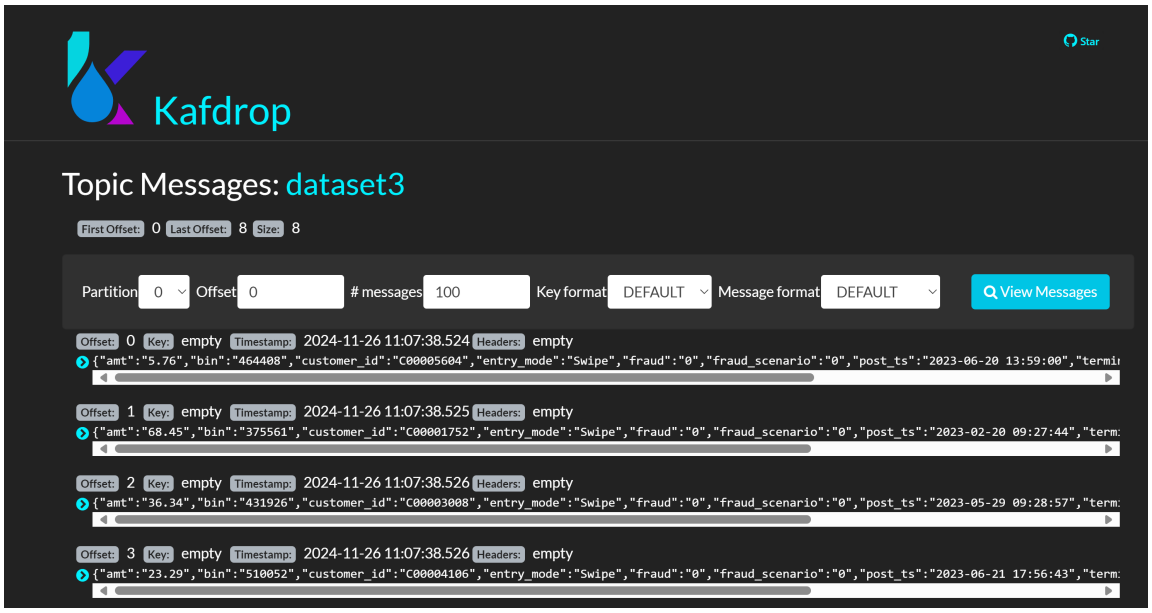
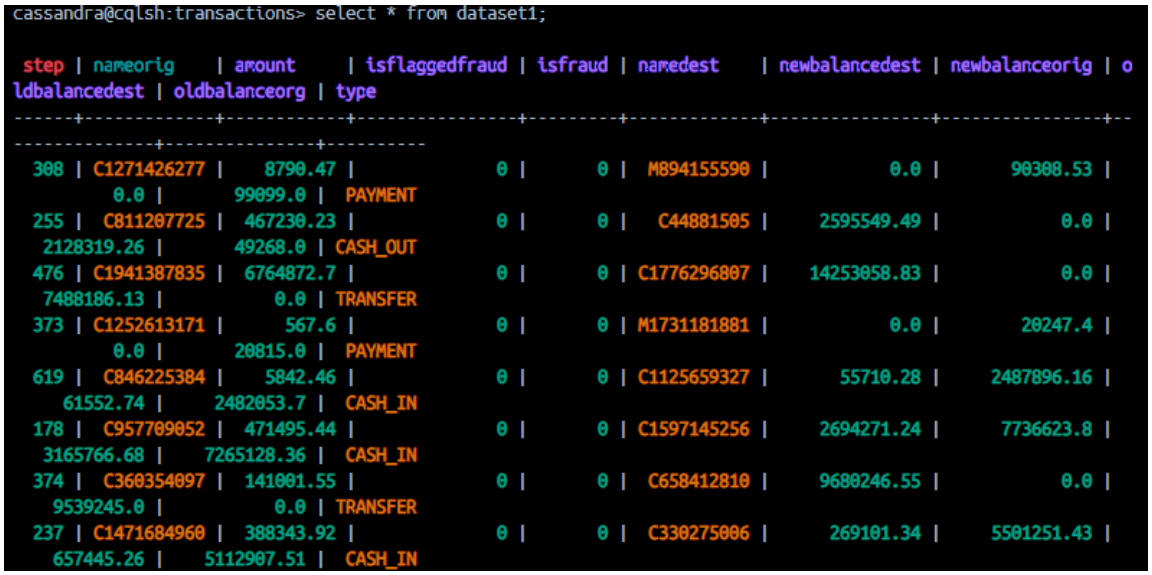


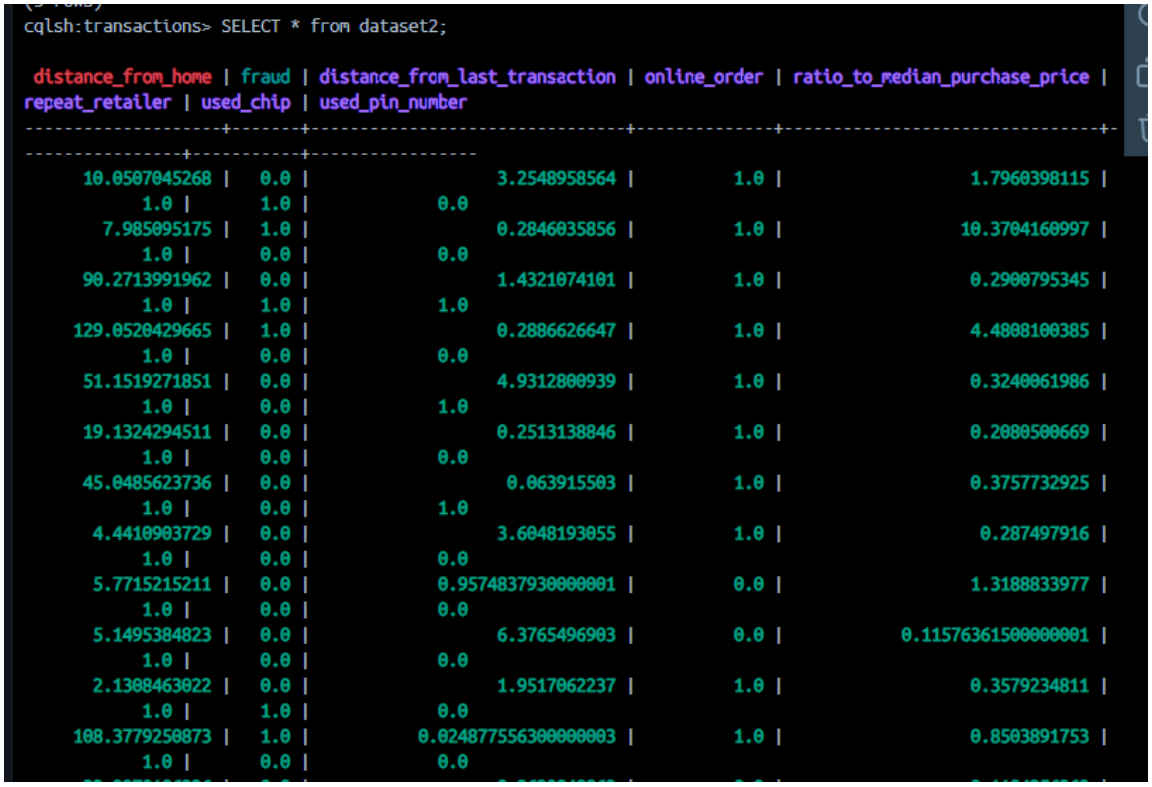
Figure 4: Kafka Dataset3



```
cassandra@cqlsh:transactions> select * from dataset1;

 step | nameorig    | amount     | isflaggedfraud | isfraud | namedest    | newbalancedest | newbalanceorig | o
ldbalancedest | oldbalanceorg | type
------+-------------+------------+----------------+---------+-------------+----------------+----------------+--
--------------+---------------+----------
  308 | C1271426277 |    8790.47 |              0 |       0 |  M894155590 |            0.0 |       90308.53 |
          0.0 |       99099.0 |  PAYMENT
  255 |  C811207725 |  467230.23 |              0 |       0 |   C44881505 |     2595549.49 |            0.0 |
   2128319.26 |       49268.0 | CASH_OUT
  476 | C1941387835 |  6764872.7 |              0 |       0 | C1776296807 |    14253058.83 |            0.0 |
    7488186.13 |           0.0 | TRANSFER
  373 | C1252613171 |      567.6 |              0 |       0 | M1731181881 |            0.0 |        20247.4 |
          0.0 |       20815.0 |  PAYMENT
  619 |  C846225384 |    5842.46 |              0 |       0 | C1125659327 |       55710.28 |     2487896.16 |
     61552.74 |     2482053.7 |  CASH_IN
  178 |  C957709052 |  471495.44 |              0 |       0 | C1597145256 |     2694271.24 |      7736623.8 |
   3165766.68 |    7265128.36 |  CASH_IN
  374 |  C360354097 |  141001.55 |              0 |       0 |  C658412810 |     9680246.55 |            0.0 |
    9539245.0 |           0.0 | TRANSFER
  237 | C1471684960 |  388343.92 |              0 |       0 |  C330275006 |      269101.34 |     5501251.43 |
    657445.26 |    5112907.51 |  CASH_IN
```

Figure 5: Cassandra Dataset1



```
cqlsh:transactions> SELECT * from dataset2;

 distance_from_home | fraud | distance_from_last_transaction | online_order | ratio_to_median_purchase_price |
repeat_retailer | used_chip | used_pin_number
--------------------+-------+--------------------------------+--------------+--------------------------------+-
----------------+-----------+-----------------
      10.0507045268 |   0.0 |                 3.2548958564 |          1.0 |                  1.7960398115 |
            1.0 |       1.0 |              0.0
        7.985095175 |   1.0 |                 0.2846035856 |          1.0 |                 10.3704160997 |
            1.0 |       0.0 |              0.0
      90.2713991962 |   0.0 |                 1.4321074101 |          1.0 |                  0.2900795345 |
            1.0 |       1.0 |              1.0
     129.0520429665 |   1.0 |                 0.2886626647 |          1.0 |                  4.4808100385 |
            1.0 |       0.0 |              0.0
      51.1519271851 |   0.0 |                 4.9312800939 |          1.0 |                  0.3240061986 |
            1.0 |       0.0 |              1.0
      19.1324294511 |   0.0 |                 0.2513138846 |          1.0 |                  0.2080500669 |
            1.0 |       0.0 |              0.0
      45.0485623736 |   0.0 |                  0.063915503 |          1.0 |                  0.3757732925 |
            1.0 |       0.0 |              1.0
       4.4410903729 |   0.0 |                 3.6048193055 |          1.0 |                   0.287497916 |
            1.0 |       0.0 |              0.0
       5.7715215211 |   0.0 |              0.9574837930000001 |          0.0 |                  1.3188833977 |
            1.0 |       0.0 |              0.0
       5.1495384823 |   0.0 |                 6.3765496903 |          0.0 |           0.11576361500000001 |
            1.0 |       0.0 |              0.0
       2.1308463022 |   0.0 |                 1.9517062237 |          1.0 |                  0.3579234811 |
            1.0 |       1.0 |              0.0
     108.3779250873 |   1.0 |           0.024877556300000003 |          1.0 |                  0.8503891753 |
            1.0 |       0.0 |              0.0
```

Figure 6: Cassandra Dataset2

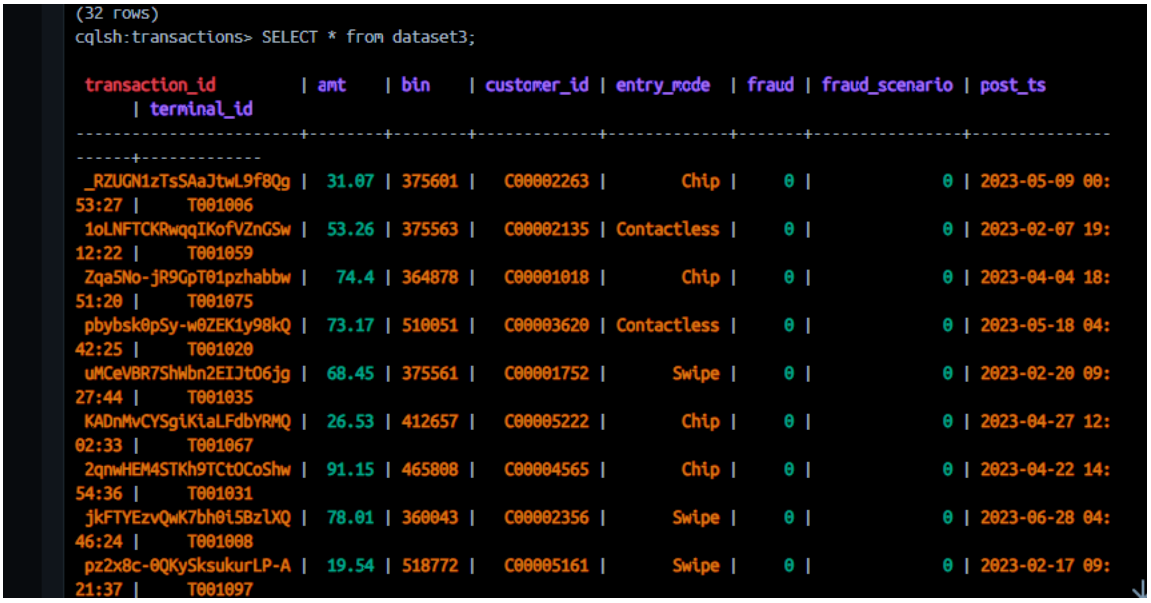


Figure 7: Cassandra Dataset3

Test objective	**Steps**	**Expected Result**	**Actual Result**
Verify correct data preprocessing of dataset 1	Run 'pytest' from the root folder	Feature 'type' is correctly transformed into numeric value (5 cases); Feature 'isMerchant' is correctly prepared (2 cases)	PASSED. The screenshot is provided in Fig. 3
Verify correct data preprocessing 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 Fig. 3
Verify correct data preprocessing 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 Fig. 3
Verify correct data preprocessing of dataset 4	Run 'pytest' from the root folder	Features 'Amount', 'Class' are renamed to 'amount' and 'isFraud'; Extra features are removed	PASSED. The screenshot is provided in Fig. 3

Table 2: Data pre-processing tests

```
============================================= 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 8: Unit testing result

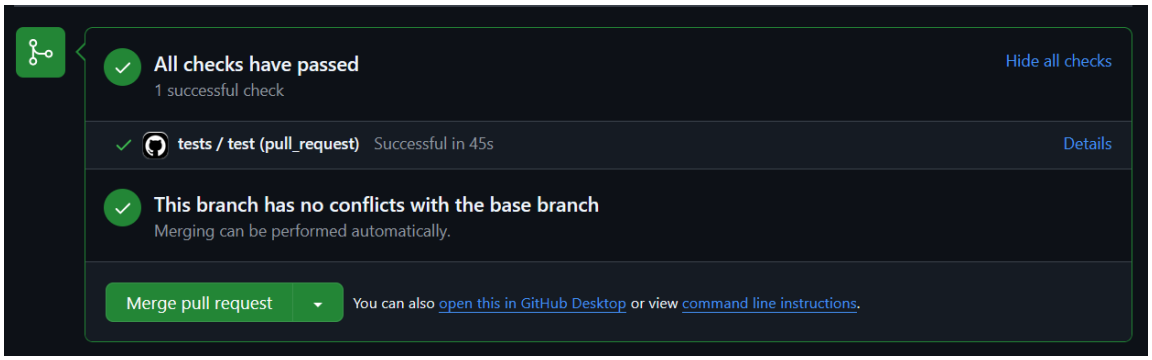


Figure 9: GitHub checks before merge