

PSD

Projekt – raport

Bartłomiej Baran 304007
Michał Kaczmarczyk 305123

1 Opis projektu

Celem projektu było opracowanie oprogramowania służącego do wykrywania anomalii w transakcjach dokonywanych kartami płatniczymi przez ich użytkowników.

2 Architektura projektu

2.1 Schemat architektury

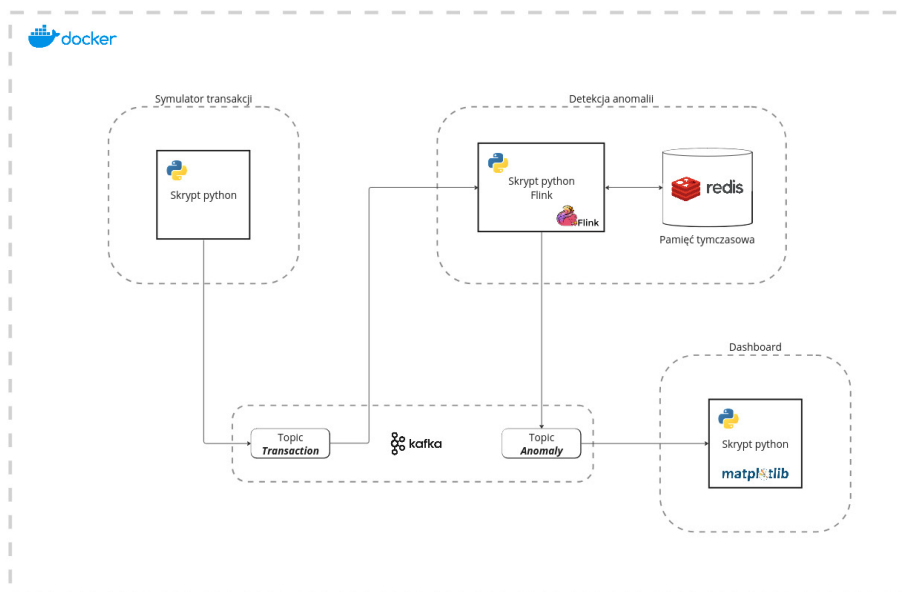


Figure 1: Architektura oprogramowania

2.2 Komponenty

Aplikacja składa się z następujących komponentów, każdy z komponentów został skonteneryzowany:

- **Zookeeper** - wykorzystywany do zarządzania klastrami Kafki. Zookeeper koordynuje i synchronizuje pracę brokerów Kafki, zapewniając stabilność i niezawodność przesyłania strumieni danych.
- **Kafka** - służy do przesyłania strumieni danych między różnymi komponentami aplikacji. Transakcje są przesyłane do Kafki, skąd są następnie konsumowane przez Flink do dalszego przetwarzania.
- **Kafdrop** - umożliwia wizualizację i zarządzanie topicami w Kafce, co ułatwia monitorowanie przesyłanych danych.
- **Skrypty** - skrypty napisane w Pythonie:
 - symulator transakcji
 - detektor anomalii, korzystający z narzędzia Flink
 - wizualizator wykrytych anomalii, korzystający z biblioteki *matplotlib*
- **Redis** - wykorzystywany jako pamięć tymczasowa do przechowywania danych potrzebnych do detekcji anomalii. Redis zapewnia szybki dostęp do tych danych, co jest kluczowe dla wydajności przetwarzania danych przez Flinka.

3 Oprogramowanie

3.1 Symulator

Do generowania transakcji kart płatniczych przygotowany został symulator. Transakcje generowane są dla 10000 różnych kart, które przypisane są do 7000 użytkowników.

Struktura danych, które generowane są przez symulator:

```
1 {  
2     "card_id": <id karty platniczej>,  
3     "user_id": <id uzytkownika>,  
4     "latitude": <szerokosc geograficzna ostatniej  
5         transakcji>,  
6     "longitude": <dlugosc geograficzna ostatniej  
7         transakcji>,  
8     "value": <wartosc transakcji>,  
9     "limit": <limit transakcji>,  
10    "timestamp": <stempel czasowy transakcji>,  
11 }
```

Przed rozpoczęciem generowania transakcji symulator inicjalizuje trzy mapy:

1. Mapa $card_id \rightarrow user_id$ – mapa trzymająca $user_id$ właścicieli kart płatniczych, inicjalizowana losowym przydziałem właścicieli
2. Mapa $user_id \rightarrow last_location$ – mapa przechowująca informacje na temat miejsca (współrzędnych) ostatniej transakcji, inicjalizacja losowymi współrzędnymi geograficznymi
3. Mapa $card_id \rightarrow limit$ – mapa przechowująca informacje na temat limitów każdej z kart płatniczych, inicjalizacja losową wartością z rozkładu normalnego (20000, 40000)

Generowanie danych transakcji odbywa się w następujący sposób:

- $card_id$ – wartość całkowitoliczbowa losowana z przedziału $[0; 10000)$
- $user_id$ – wartość wyciągana z mapy przechowującej właścicieli kart płatniczych
- $latitude$ – losowa szerokość geograficzna w promieniu 100 km od punktu, którym jest ostatnia lokalizacja użytkownika wyciągnięta z mapy przechowującej lokalizacje ostatnich transakcji użytkowników
- $longitude$ – losowa długość geograficzna w promieniu 100 km od punktu, którym jest ostatnia lokalizacja użytkownika wyciągnięta z mapy przechowującej lokalizacje ostatnich transakcji użytkowników
- $value$ – wartość transakcji, losowana z rozkładu normalnego o parametrach $(limit/2; limit/2)$
- $limit$ – limit transakcji karty wyciągany z mapy przechowującej limity kart płatniczych
- $timestamp$ – aktualny stempel czasowy

Transakcje generowane są w losowych odstępach czasu od 1 do 5 sekund i wysyłane są na topic Kafki o nazwie ***Transaction***

3.2 Detektor anomalii

Detekcja anomalii odbywa się za pomocą dedykowanego komponentu, który dodatkowo pełni rolę konsumenta Kafka czytującego transakcje znajdujące się na topicu ***Transaction***. Wykryte anomalie wysyłane są oddzielnym topicem o nazwie ***Anomaly*** ich struktura jest analogiczna do struktury danych przesyłanych przez topic ***Transaction***, jest jednak dodatkowo rozszerzona o pole *anomaly*, która zawiera wskazanie, która anomalia została wykryta.

3.2.1 Anomalie

W ramach detektora anomalii zaimplementowane zostały cztery typy anomalii:

1. Szybka zmiana lokalizacji – użytkownik zmienił lokalizację z prędkością większą niż $900 \frac{km}{h}$ (typowa prędkość samolotu pasażerskiego), do wykrywania tego typu anomalii wykorzystana została baza Redis, w której przechowywana jest ostatnia lokalizacja każdego użytkownika.
2. Znacząco odstająca wartość transakcji – wartość transakcji przekroczyła dwukrotnie średnią wartość transakcji danego użytkownika, do przechowywania średniej wartości transakcji również wykorzystana została baza Redis
3. Zbyt wysoka częstotliwość transakcji – użytkownik wykonał więcej niż cztery transakcje w jednominutowym oknie czasowym
4. Trzykrotna próba transakcji przekraczającej limit – użytkownik na danej karcie płatniczej wykonał więcej niż trzy transakcje przekraczające limit

4 Testy i wyniki

Podczas testów systemu dokonano zmniejszenia liczby użytkowników, co miało kluczowe znaczenie dla optymalizacji procesu wykrywania anomalii. Dzięki temu ograniczeniu możliwe jest zwiększenie częstotliwości występowania anomalii, co przekłada się na szybszą identyfikację ewentualnych problemów w działaniu systemu. Generator transakcji operuje na podstawie rozkładu normalnego, losując wartości transakcji z określonym prawdopodobieństwem. Gdy wygenerowana transakcja spełnia kryteria bycia anomalia, jest natychmiast raportowana i przekazywana do odpowiedniego kanału. Dodatkowo, w interfejsie graficznym użytkownicy mają dostęp do tabeli prezentującej ostatnie 5 transakcji, gdzie kolorem czerwonym oznaczone są te podejrzane, a zielonym te uznane za zwyczajne. Ponadto, na dwóch wykresach słupkowych zebrane są dane dotyczące częstości występowania poszczególnych rodzajów anomalii, a na wykresie zbiorczym można śledzić ilości transakcji prawidłowych oraz transakcji uznanych za podejrzane. Ta pełna i przejrzysta prezentacja danych pozwala użytkownikom skutecznie monitorować działanie systemu i szybko reagować na ewentualne nieprawidłowości. Komponenty architektury były testowane w trakcie rozwijania projektu, na poniższych zrzutach ekranu zostały przedstawione logi oraz zrzut ekranu z dashboardu.

```

[+] Running 9/9
✔ Network psd-default          Created           0.1s
✔ Container psd-zookeeper-1     Created           0.0s
✔ Container psd-redis-1         Created           0.0s
✔ Container psd-kafka-1         Created           0.1s
✔ Container psd-flink-jobmanager-1 Created           0.0s
✔ Container psd-python-app-1    Created           0.0s
✔ Container psd-kafdrop-1       Created           0.0s
✔ Container psd-flink-taskmanager-1 Created           0.0s
Attaching to flink-jobmanager-1, flink-taskmanager-1, kafdrop-1, kafka-1, python-app-1, redis-1, zookeeper-1

```

Figure 2: Uruchomienie kontenerów

```

flink-taskmanager-1 | 2024-06-10 09:22:18,712 INFO org.apache.flink.runtime.filesystem.FileCache | User file cache uses d
Directory /tmp/flink-dist-cache-83306062-c872-45cd-8546-5d284c624087
flink-taskmanager-1 | 2024-06-10 09:22:18,514 INFO org.apache.flink.runtime.taskexecutor.TaskExecutor | - Connecting to Resource
Manager pekko.tcp://flink@flink-jobmanager:6123/user/rpc/resourcemanager.* (00000000000000000000000000000000).
flink-jobmanager-1 | 2024-06-10 09:22:18,671 ERROR org.apache.pekko.remote.EndpointWriter | - dropping message [clas
s org.apache.pekko.actor.ActorSelectionMessage] for non-local recipient [Actor[pekko.tcp://flink@flink-jobmanager:6123/]] arriving at [pekko.
tcp://flink@flink-jobmanager:6123] inbound addresses are [pekko.tcp://flink@53ecbf4e0d52:6123]
kafka-1 | 2024-06-10 09:22:22,095 INFO [Controller id=1] Processing automatic preferred replica leader election (kafka.controller.Kafk
aController)
kafka-1 | 2024-06-10 09:22:22,096 TRACE [Controller id=1] Checking need to trigger auto leader balancing (kafka.controller.Kaf
kaController)
kafka-1 | 2024-06-10 09:22:22,098 DEBUG [Controller id=1] Topics not in preferred replica for broker 1 Map() (kafka.controller.Kafk
aController)
kafka-1 | 2024-06-10 09:22:22,099 TRACE [Controller id=1] Leader imbalance ratio for broker 1 is 0.0 (kafka.controller.KafkaCo
ntroller)
flink-taskmanager-1 | 2024-06-10 09:22:28,541 INFO org.apache.flink.runtime.taskexecutor.TaskExecutor | - Could not resolve Reso
urceManager address pekko.tcp://flink@flink-jobmanager:6123/user/rpc/resourcemanager_*, retrying in 10000 ms: Could not connect to rpc endpoi
nt under address pekko.tcp://flink@flink-jobmanager:6123/user/rpc/resourcemanager_*.
flink-jobmanager-1 | 2024-06-10 09:22:38,559 ERROR org.apache.pekko.remote.EndpointWriter | - dropping message [clas
s org.apache.pekko.actor.ActorSelectionMessage] for non-local recipient [Actor[pekko.tcp://flink@flink-jobmanager:6123/]] arriving at [pekko.
tcp://flink@flink-jobmanager:6123] inbound addresses are [pekko.tcp://flink@53ecbf4e0d52:6123]
flink-taskmanager-1 | 2024-06-10 09:22:48,576 INFO org.apache.flink.runtime.taskexecutor.TaskExecutor | - Could not resolve Reso
urceManager address pekko.tcp://flink@flink-jobmanager:6123/user/rpc/resourcemanager_*, retrying in 10000 ms: Could not connect to rpc endpoi
nt under address pekko.tcp://flink@flink-jobmanager:6123/user/rpc/resourcemanager_*.
flink-jobmanager-1 | 2024-06-10 09:22:58,599 ERROR org.apache.pekko.remote.EndpointWriter | - dropping message [clas
s org.apache.pekko.actor.ActorSelectionMessage] for non-local recipient [Actor[pekko.tcp://flink@flink-jobmanager:6123/]] arriving at [pekko.
tcp://flink@flink-jobmanager:6123] inbound addresses are [pekko.tcp://flink@53ecbf4e0d52:6123]
flink-taskmanager-1 | 2024-06-10 09:23:08,617 INFO org.apache.flink.runtime.taskexecutor.TaskExecutor | - Could not resolve Reso
urceManager address pekko.tcp://flink@flink-jobmanager:6123/user/rpc/resourcemanager_*, retrying in 10000 ms: Could not connect to rpc endpoi
nt under address pekko.tcp://flink@flink-jobmanager:6123/user/rpc/resourcemanager_*.
flink-jobmanager-1 | 2024-06-10 09:23:18,639 ERROR org.apache.pekko.remote.EndpointWriter | - dropping message [clas
s org.apache.pekko.actor.ActorSelectionMessage] for non-local recipient [Actor[pekko.tcp://flink@flink-jobmanager:6123/]] arriving at [pekko.
tcp://flink@flink-jobmanager:6123] inbound addresses are [pekko.tcp://flink@53ecbf4e0d52:6123]
flink-taskmanager-1 | 2024-06-10 09:23:28,656 INFO org.apache.flink.runtime.taskexecutor.TaskExecutor | - Could not resolve Reso
urceManager address pekko.tcp://flink@flink-jobmanager:6123/user/rpc/resourcemanager_*, retrying in 10000 ms: Could not connect to rpc endpoi
nt under address pekko.tcp://flink@flink-jobmanager:6123/user/rpc/resourcemanager_*.

```

Figure 3: Uruchomione kontenery

```

python-app-1 | Transaction | Details = {'card_id': 2, 'user_id': 0, 'latitude': -3.690123637836063, 'longitude': -79.84704232465528,
'value': 3041.6815615751084, 'limit': 50552.46069754833, 'timestamp': '2024-06-10 09:23:26'}
python-app-1 | Transaction | Details = {'card_id': 2, 'user_id': 0, 'latitude': -3.6899963524012804, 'longitude': -79.84751644112166,
'value': 30712.562122013376, 'limit': 50552.46069754833, 'timestamp': '2024-06-10 09:23:29'}
python-app-1 | Transaction | Details = {'card_id': 0, 'user_id': 0, 'latitude': -3.6897195963713854, 'longitude': -79.84830618630902,
'value': 4941.502290856729, 'limit': 5214.863661034566, 'timestamp': '2024-06-10 09:23:31'}
python-app-1 | Transaction | Details = {'card_id': 1, 'user_id': 2, 'latitude': 18.1845085807606907, 'longitude': -99.79543583841007,
'value': 19871.34311626136, 'limit': 27790.56522470434, 'timestamp': '2024-06-10 09:23:35'}
python-app-1 | Transaction | Details = {'card_id': 0, 'user_id': 0, 'latitude': -3.6905818869774425, 'longitude': -79.84725482287766,
'value': 4008.6499792133945, 'limit': 5214.863661034566, 'timestamp': '2024-06-10 09:23:39'}
python-app-1 | Transaction | Details = {'card_id': 1, 'user_id': 2, 'latitude': 18.185661244105408, 'longitude': -99.79548836147626,
'value': 11323.679933065981, 'limit': 27790.56522470434, 'timestamp': '2024-06-10 09:23:42'}
python-app-1 | Transaction | Details = {'card_id': 1, 'user_id': 2, 'latitude': 18.184864244624585, 'longitude': -99.7949362907985,
'value': 19277.319943619677, 'limit': 27790.56522470434, 'timestamp': '2024-06-10 09:23:46'}
python-app-1 | Transaction | Details = {'card_id': 1, 'user_id': 2, 'latitude': 18.185137089477507, 'longitude': -99.79595512743221,
'value': 22972.686313473816, 'limit': 27790.56522470434, 'timestamp': '2024-06-10 09:23:50'}
python-app-1 | Transaction | Details = {'card_id': 1, 'user_id': 2, 'latitude': 18.18558734926879, 'longitude': -99.79536255681349,
'value': 25885.640906488818, 'limit': 27790.56522470434, 'timestamp': '2024-06-10 09:23:52'}
python-app-1 | Transaction | Details = {'card_id': 2, 'user_id': 0, 'latitude': -3.690091415249217, 'longitude': -79.84835309424243,
'value': 32470.701864181163, 'limit': 50552.46069754833, 'timestamp': '2024-06-10 09:23:56'}
python-app-1 | Transaction | Details = {'card_id': 1, 'user_id': 2, 'latitude': 18.185759152703277, 'longitude': -99.79538938366059,
'value': 32270.579375860905, 'limit': 27790.56522470434, 'timestamp': '2024-06-10 09:23:58'}
python-app-1 | Transaction | Details = {'card_id': 1, 'user_id': 2, 'latitude': 18.1859536116730376, 'longitude': -79.84704168720184,
'value': 5240.8109797318741, 'limit': 5214.863661034566, 'timestamp': '2024-06-10 09:23:59'}
python-app-1 | Transaction | Details = {'card_id': 0, 'user_id': 0, 'latitude': -3.690150729358256, 'longitude': -79.84774951603477,
'value': 6706.538939212965, 'limit': 5214.863661034566, 'timestamp': '2024-06-10 09:24:00'}
python-app-1 | Transaction | Details = {'card_id': 1, 'user_id': 2, 'latitude': 18.1848619241471, 'longitude': -99.79454895842379,
'value': 53105.40285247975, 'limit': 27790.56522470434, 'timestamp': '2024-06-10 09:24:03'}
python-app-1 | Transaction | Details = {'card_id': 0, 'user_id': 0, 'latitude': -3.6896203986099585, 'longitude': -79.84715555086653,
'value': 11085.160517940467, 'limit': 5214.863661034566, 'timestamp': '2024-06-10 09:24:08'}
python-app-1 | Transaction | Details = {'card_id': 1, 'user_id': 2, 'latitude': 18.185055146737866, 'longitude': -99.7961534769599,
'value': 19790.1119927232, 'limit': 27790.56522470434, 'timestamp': '2024-06-10 09:24:10'}
python-app-1 | Transaction | Details = {'card_id': 1, 'user_id': 2, 'latitude': 18.184437483534136, 'longitude': -99.79557140951752,
'value': 20529.444258595897, 'limit': 27790.56522470434, 'timestamp': '2024-06-10 09:24:12'}
[link-jobmanager-1 | 2024-06-10 09:24:18,759 ERROR org.apache.pekko.remote.EndpointWriter |] - dropping message [clas
s org.apache.pekko.actor.ActorSelectionMessage] for non-local recipient [Actor[pekko.tcp://flink@flink-jobmanager:6123/]] arriving at [pekko.
tcp://flink@flink-jobmanager:6123] inbound addresses are [pekko.tcp://flink@53ecbf4e0d52:6123]

```

Figure 4: Działanie generatora transakcji

```

MichalMichal-Lenovo-Legion-5-15IMH05H:~/Desktop/projects/Python/psd$ docker exec -ti psd-python-app-1 bash
root@df0217bc0c94:/app# python flink.py
+open of DetectLocationChange
Connected to Redis at redis:6379
+open of DetectTransactionValueChange
+[[1,2,18.185552297240875,-99.79501686419324,28399.831337710653,27790.56522470434,2024-06-10 09:25:16,None]
====[START] Location change anomaly detection started=====
Processing transaction for card_id=1, user_id=2
====[START] Transaction value change anomaly detection started=====
Processing transaction for user_id=2
New average transaction value for 2: 14199.91566855327
Anomaly detected: Significant increase in transaction value
====[END] Transaction value change anomaly detection started=====
Saving last location of user 2
====[END] Location change anomaly detection =====
+[[1,2,18.1853297921125,-99.79587112142283,29208.48843756166,27790.56522470434,2024-06-10 09:25:21,None]
====[START] Location change anomaly detection started=====
Processing transaction for card_id=1, user_id=2
Calculated speed for user_id=2 is 67.37372161247823 km/h
Saving last location of user 2
====[START] Transaction value change anomaly detection started=====
Processing transaction for user_id=2
====[END] Location change anomaly detection =====
New average transaction value for 2: 21704.202053208493
Anomaly detected: Significant increase in transaction value
====[END] Transaction value change anomaly detection started=====
+[[0,0,-3.6898402242669306,-79.84700724583212,2942.9828567553564,5214.863661034566,2024-06-10 09:25:23,None]
====[START] Transaction value change anomaly detection started=====
Processing transaction for user_id=0
====[START] Location change anomaly detection started=====
Processing transaction for card_id=0, user_id=0
New average transaction value for 0: 1471.4914283776782
Anomaly detected: Significant increase in transaction value
====[END] Transaction value change anomaly detection started=====
Saving last location of user 0
====[END] Location change anomaly detection =====

```

Figure 5: Przetwarzanie flink

```

michał@michał-Lenovo-Legion-5-15IMH05H:~/Desktop/projects/Python/psd$ python new_dash.py
/home/michał/Desktop/projects/Python/psd/new_dash.py:62: UserWarning: frames=None which we can infer the length of, did not pass an explicit
'save_count' and passed cache_frame_data=True. To avoid a possibly unbounded cache, frame data caching has been disabled. To suppress this w
arning either pass 'cache_frame_data=False' or 'save_count=MAX_FRAMES'.
  self ani = animation.FuncAnimation(self.fig, self.update_plots, interval=1000)
Processing message from topic Transaction: {'card_id': 2, 'user_id': 0, 'latitude': -3.6904250090846706, 'longitude': -79.84825971584662, 'va
lue': 68261.90581066671, 'limit': 50552.46069754833, 'timestamp': '2024-06-10 09:22:18'}
Processing message from topic Transaction: {'card_id': 2, 'user_id': 0, 'latitude': -3.690330759062525, 'longitude': -79.84713251664182, 'va
lue': 53334.81686125859, 'limit': 50552.46069754833, 'timestamp': '2024-06-10 09:22:22'}
Processing message from topic Transaction: {'card_id': 2, 'user_id': 0, 'latitude': -3.689297242732023, 'longitude': -79.84778837720691, 'va
lue': 39410.940127704256, 'limit': 50552.46069754833, 'timestamp': '2024-06-10 09:22:27'}
Processing message from topic Transaction: {'card_id': 2, 'user_id': 0, 'latitude': -3.690048076612539, 'longitude': -79.84837963018288, 'va
lue': 78963.88857937651, 'limit': 50552.46069754833, 'timestamp': '2024-06-10 09:22:28'}
Processing message from topic Transaction: {'card_id': 0, 'user_id': 0, 'latitude': -3.6896634558174637, 'longitude': -79.84751331925578, 'va
lue': 2390.5278812050433, 'limit': 5214.863661034566, 'timestamp': '2024-06-10 09:22:32'}
Processing message from topic Transaction: {'card_id': 0, 'user_id': 0, 'latitude': -3.68948494073524, 'longitude': -79.84767052559313, 'valu
e': 4649.254407850709, 'limit': 5214.863661034566, 'timestamp': '2024-06-10 09:22:36'}
Processing message from topic Transaction: {'card_id': 0, 'user_id': 0, 'latitude': -3.690644910601996, 'longitude': -79.8475003402112, 'valu
e': 5765.805284811836, 'limit': 5214.863661034566, 'timestamp': '2024-06-10 09:22:37'}
Processing message from topic Transaction: {'card_id': 2, 'user_id': 0, 'latitude': -3.6901974897775323, 'longitude': -79.84750006089986, 'va
lue': 37824.08340819103, 'limit': 50552.46069754833, 'timestamp': '2024-06-10 09:22:41'}
Processing message from topic Transaction: {'card_id': 1, 'user_id': 2, 'latitude': 18.185735881059834, 'longitude': -99.79482014988628, 'va
lue': 34076.71714608034, 'limit': 27790.56522470434, 'timestamp': '2024-06-10 09:22:45'}
Processing message from topic Transaction: {'card_id': 2, 'user_id': 0, 'latitude': -3.6904698853966313, 'longitude': -79.84821021155378, 'va
lue': 86669.40565659417, 'limit': 50552.46069754833, 'timestamp': '2024-06-10 09:22:50'}
Processing message from topic Transaction: {'card_id': 1, 'user_id': 2, 'latitude': 18.184905261638207, 'longitude': -99.79616267875815, 'va
lue': 32021.70259511716, 'limit': 27790.56522470434, 'timestamp': '2024-06-10 09:22:53'}
Processing message from topic Transaction: {'card_id': 2, 'user_id': 0, 'latitude': -3.689943009209064, 'longitude': -79.84731920379515, 'va
lue': 7758.1588465063105, 'limit': 50552.46069754833, 'timestamp': '2024-06-10 09:22:54'}
Processing message from topic Transaction: {'card_id': 0, 'user_id': 0, 'latitude': -3.6897127441260014, 'longitude': -79.84674005663273, 'va
lue': 4511.432724837725, 'limit': 5214.863661034566, 'timestamp': '2024-06-10 09:22:55'}
Processing message from topic Transaction: {'card_id': 0, 'user_id': 0, 'latitude': -3.689923969919053, 'longitude': -79.84755972665269, 'va
lue': 12953.782577967966, 'limit': 5214.863661034566, 'timestamp': '2024-06-10 09:22:58'}
Processing message from topic Transaction: {'card_id': 1, 'user_id': 2, 'latitude': 18.184533787033665, 'longitude': -99.79529683476336, 'va
lue': 40996.596801165986, 'limit': 27790.56522470434, 'timestamp': '2024-06-10 09:23:01'}
Processing message from topic Transaction: {'card_id': 1, 'user_id': 2, 'latitude': 18.18435504575697, 'longitude': -99.79511130522013, 'valu
e': 50496.09257401183, 'limit': 27790.56522470434, 'timestamp': '2024-06-10 09:23:03'}

```

Figure 6: Dane odbierane przez panel do wizualizacji

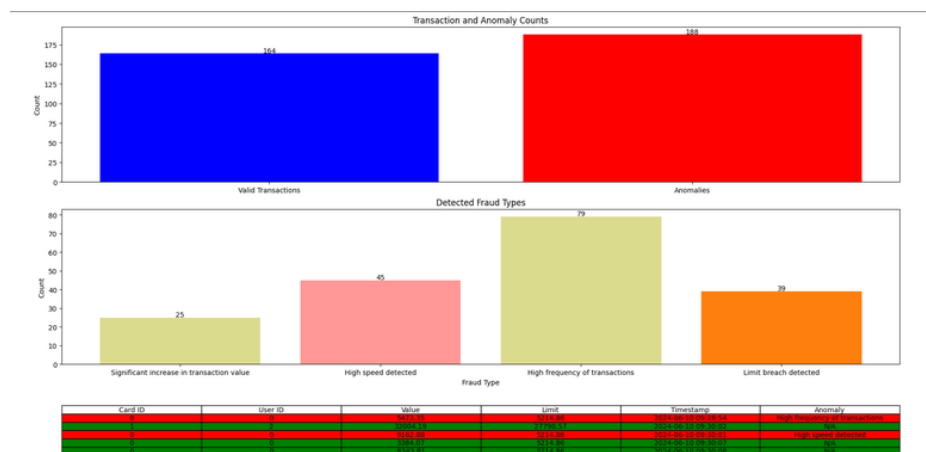


Figure 7: Zrzut ekranu z monitoringu