# ДОДАТОК А

## Вихідний код програмної моделі

```
"""# Imports"""

import pandas as pd
import numpy as np
import warnings
from matplotlib import pyplot
from datetime import datetime
from sklearn.preprocessing import MinMaxScaler
from sklearn.cluster import DBSCAN
from tqdm.notebook import tqdm

"""# Constants and functions"""

drive_path = "Anomaly Detection/"

fields = ['datetime',
          'int_datetime',
          'date',
          'int_date',
          'time',
          'int_time',
          'user_id',
          'device_id',
          'IP_id',
          'latitude',
          'longitude']

aux_fields = [
              'datetime',
```

```
                    'int_datetime',
                    'date',
                    'int_date',
                    'time',
                    'int_time',
                    'user_id',
                    'device_id',
                    'IP_id',
                    'latitude',
                    'longitude'
    ]

    def clusterData(data):
        # Get features to train
        X = data.drop(aux_fields, axis=1)
        clustering_model = DBSCAN(eps=2, min_samples=2,
metric='euclidean', leaf_size=30)
        clustering_model.fit(X)
        data['cluster'] = clustering_model.labels_
        # Return processed data and outliers list
        return data, data[data['cluster'] == -1]

    def preprocessData(data):
        # Categorize columns
        data['username'] = pd.Categorical(data['username'])
        data['user_id'] = data['username'].cat.codes
        data['device_uuid'] =
pd.Categorical(data['device_uuid'])
        data['device_id'] = data['device_uuid'].cat.codes
        data['IP_address'] = pd.Categorical(data['IP_address'])
        data['IP_id'] = data['IP_address'].cat.codes

        # Extracting date and time features
```

```
        data['datetime'] = pd.to_datetime(data['datetime'])
        data['date'] = data['datetime'].dt.date
        # Get the date as day's number
        min_date = data['date'].min()
        data['int_date'] = [(row['date'] - min_date).days for i,
row in data.iterrows()]
        data['time'] = data['datetime'].dt.time
        # Get the time in microseconds
        data['int_time'] = [row['time'].microsecond
                            + row['time'].second*1000000
                            + row['time'].minute*60*1000000
                            + row['time'].hour*3600*1000000
                            for i, row in data.iterrows()]
        # Get the datetime in microseconds
        data['int_datetime'] = data['int_time'] +
86400*1000000*data['int_date']

        return data

    def normalizeData(data):

        # datetime normalization
        data['norm_datetime'] = data['int_datetime'] / (12 *
3600 * 1000000)

        # Coordinations normalization
        data['norm_latitude'] = MinMaxScaler(feature_range=(0,
(data['latitude'].max()

- data['latitude'].min()) * 10 + 1)) \
            .fit_transform(df[['latitude']])
        data['norm_longitude'] = MinMaxScaler(feature_range=(0,
(data['longitude'].max()
```

```
- data['longitude'].min()) * 10 + 1)) \
            .fit_transform(df[['longitude']])

        return data

    if __name__ = "__main__":

        # Download and preprocess

preprocessData(pd.read_csv(drive_path+'dataset.csv').sort_values(
by=['datetime']))[fields].to_csv(drive_path+"prepared_data.csv",
index=False)

        # Normalize data
        df =
normalizeData(pd.read_csv(drive_path+"prepared_data.csv"))

        # Users analysis
        users =
pd.DataFrame(df['user_id'].value_counts().reset_index()).rename(c
olumns={"user_id": "count", "index": "user_id"})
        users['transactions'] = [df[df['user_id'] ==
row['user_id']].reset_index(drop=True) for i, row in
users.iterrows()]
        users['anomalies'] = [pd.DataFrame() for i, row in
users.iterrows()]
        users['has_anomalies'] = False
        users['device_count'] =
[len(row['transactions']['device_id'].value_counts()) for i, row
in users.iterrows()]
```

```
        users['IP_count'] =
[len(row['transactions']['IP_id'].value_counts()) for i, row in
users.iterrows()]

        # Analyze only users with more than 1 device and IP
address
        users_to_analyze = users[(users['device_count'] > 1) &
(users['IP_count'] > 1)]

        for i, user in tqdm(users_to_analyze.iterrows(),
total=len(users_to_analyze)):
            try:
                local_df = user['transactions'].copy()
                # Split device_id and IP_id columns to boolean
features
                local_df =
local_df.join(pd.get_dummies(local_df['device_id'],
prefix="device"))
                local_df =
local_df.join(pd.get_dummies(local_df['IP_id'], prefix="IP"))
                # Use clustering to find anomalies
                local_df, anomalies = clusterData(local_df)
                users['transactions'][i] = local_df
                if len(anomalies) > 0:
                    users['anomalies'][i] = anomalies
                    users['has_anomalies'][i] = True
            except Exception as e:
                print("Exception: " + str(e))

        # Write results to CSV

users[users['has_anomalies']][aux_fields].to_csv(drive_path+"susp
icious_users.csv")
```

```
        # Devices analysis
        devices =
pd.DataFrame(df['device_id'].value_counts().reset_index()).rename
(columns={"device_id": "count", "index": "device_id"})
        devices['transactions'] = [df[df['device_id'] ==
row['device_id']].reset_index(drop=True) for i, row in
devices.iterrows()]
        devices['anomalies'] = [pd.DataFrame() for i, row in
devices.iterrows()]
        devices['has_anomalies'] = False
        devices['user_count'] =
[len(row['transactions']['user_id'].value_counts()) for i, row in
devices.iterrows()]
        devices['IP_count'] =
[len(row['transactions']['IP_id'].value_counts()) for i, row in
devices.iterrows()]

        # Analyze only users with more than 1 IP address
        devices_to_analyze = devices[(devices['IP_count'] > 1)]

        for i, device in tqdm(devices_to_analyze.iterrows(),
total=len(devices_to_analyze)):
            try:
                local_df = device['transactions'].copy()
                # Split device_id and IP_id columns to boolean
features
                local_df =
local_df.join(pd.get_dummies(local_df['user_id'], prefix="user"))
                local_df =
local_df.join(pd.get_dummies(local_df['IP_id'], prefix="IP"))
                # Use clustering to find anomalies
                local_df, anomalies = clusterData(local_df)
```

```
                devices['transactions'][i] = local_df
                if len(anomalies) > 0:
                    devices['anomalies'][i] = anomalies
                    devices['has_anomalies'][i] = True
            except Exception as e:
                print("Exception: " + str(e))

        # Write results to CSV

devices[(devices['has_anomalies'])][aux_fields].to_csv(drive_path
+"suspicious_devices.csv")
```

# ДОДАТОК Б

## Знімки екрану роботи програми

| | datetime | user_id | device_id | IP_id | latitude | longitude |
|---|---|---|---|---|---|---|
| 0 | 2022-02-01 00:00:00.037000+00:00 | 1331 | 934 | 4732 | 40.632982 | -74.280504 |
| 1 | 2022-02-01 00:00:00.211000+00:00 | 1137 | 189 | 4137 | 40.775897 | -74.238230 |
| 2 | 2022-02-01 00:00:00.263000+00:00 | 1160 | 387 | 5140 | 39.360178 | -74.419202 |
| 3 | 2022-02-01 00:00:00.579000+00:00 | 450 | 321 | 4449 | 40.652104 | -74.237631 |
| 4 | 2022-02-01 00:00:00.739000+00:00 | 1173 | 837 | 2939 | 39.875614 | -75.130533 |
| 5 | 2022-02-01 00:00:00.830000+00:00 | 11 | 1521 | 5146 | 40.720805 | -74.042509 |
| 6 | 2022-02-01 00:00:01.082000+00:00 | 588 | 1715 | 2782 | 39.920974 | -75.109249 |
| 7 | 2022-02-01 00:00:01.182000+00:00 | 576 | 905 | 6515 | 40.784144 | -74.011237 |
| 8 | 2022-02-01 00:00:01.891000+00:00 | 116 | 1116 | 5339 | 40.777454 | -74.018305 |
| 9 | 2022-02-01 00:00:02.004000+00:00 | 448 | 1272 | 32 | 40.858649 | -74.139829 |
| 10 | 2022-02-01 00:00:02.025000+00:00 | 655 | 1258 | 7893 | 40.728382 | -74.148632 |
| 11 | 2022-02-01 00:00:02.067000+00:00 | 1420 | 402 | 3646 | 40.724552 | -74.050549 |
| 12 | 2022-02-01 00:00:02.086000+00:00 | 507 | 730 | 2846 | 39.642233 | -75.515041 |
| 13 | 2022-02-01 00:00:02.815000+00:00 | 1425 | 1247 | 9084 | 39.452898 | -74.491057 |
| 14 | 2022-02-01 00:00:02.821000+00:00 | 1187 | 637 | 5243 | 40.732744 | -74.084239 |
| 15 | 2022-02-01 00:00:03.442000+00:00 | 1108 | 1085 | 432 | 39.902169 | -75.124881 |
| 16 | 2022-02-01 00:00:04.055000+00:00 | 715 | 406 | 5020 | 40.789106 | -74.017112 |
| 17 | 2022-02-01 00:00:04.119000+00:00 | 212 | 157 | 3381 | 40.669383 | -74.114494 |
| 18 | 2022-02-01 00:00:04.220000+00:00 | 254 | 1885 | 2937 | 39.825229 | -75.277465 |
| 19 | 2022-02-01 00:00:04.418000+00:00 | 152 | 1279 | 800 | 40.663759 | -74.390510 |
| 20 | 2022-02-01 00:00:04.508000+00:00 | 296 | 1287 | 2961 | 40.012535 | -75.002602 |

| | norm_datetime | norm_latitude | norm_longitude | device_277 | device_547 | IP_5386 | IP_6696 | IP_7799 | IP_9409 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0.046140 | 141.699426 | 461.399285 | 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0.046934 | 141.699426 | 461.399285 | 1 | 0 | 0 | 1 | 0 | 0 |
| 2 | 0.046941 | 141.700594 | 461.398333 | 1 | 0 | 0 | 1 | 0 | 0 |
| 3 | 0.047635 | 141.700634 | 461.398313 | 1 | 0 | 0 | 1 | 0 | 0 |
| 4 | 0.048376 | 141.700634 | 461.398313 | 1 | 0 | 0 | 1 | 0 | 0 |
| 5 | 0.049117 | 141.700634 | 461.398313 | 1 | 0 | 0 | 1 | 0 | 0 |
| 6 | 0.049201 | 141.700634 | 461.398313 | 1 | 0 | 0 | 1 | 0 | 0 |
| 7 | 0.049559 | 141.700634 | 461.398313 | 1 | 0 | 0 | 1 | 0 | 0 |
| 8 | 0.049708 | 141.700634 | 461.398313 | 1 | 0 | 0 | 1 | 0 | 0 |
| 9 | 0.050298 | 141.700634 | 461.398313 | 1 | 0 | 0 | 1 | 0 | 0 |
| 10 | 0.050447 | 141.700634 | 461.398313 | 1 | 0 | 0 | 1 | 0 | 0 |
| 11 | 0.050643 | 141.700634 | 461.398313 | 1 | 0 | 0 | 1 | 0 | 0 |
| 12 | 0.051390 | 141.700634 | 461.398313 | 1 | 0 | 0 | 1 | 0 | 0 |
| 13 | 0.051957 | 141.700614 | 461.398263 | 1 | 0 | 0 | 1 | 0 | 0 |
| 14 | 0.052148 | 141.700634 | 461.398313 | 1 | 0 | 0 | 1 | 0 | 0 |
| 15 | 0.052567 | 141.700634 | 461.398313 | 1 | 0 | 0 | 1 | 0 | 0 |
| 16 | 0.053314 | 141.700634 | 461.398313 | 1 | 0 | 0 | 1 | 0 | 0 |
| 17 | 0.053363 | 141.700634 | 461.398313 | 1 | 0 | 0 | 1 | 0 | 0 |
| 18 | 0.054103 | 141.700634 | 461.398313 | 1 | 0 | 0 | 1 | 0 | 0 |
| 19 | 0.054484 | 141.700634 | 461.398313 | 1 | 0 | 0 | 1 | 0 | 0 |
| 20 | 0.054841 | 141.700634 | 461.398313 | 1 | 0 | 0 | 1 | 0 | 0 |

```
users[users['has_anomalies']]
```

| | user_id | count | transactions | anomalies | has_anomalies | device_count | IP_count |
|---|---|---|---|---|---|---|---|
| 4 | 1251 | 7100 | datetime int_da... | datetime int_da... | True | 2 | 4 |
| 7 | 1122 | 6508 | datetime int_da... | datetime int_datet... | True | 2 | 2 |
| 31 | 1305 | 2622 | datetime int_da... | datetime int_datet... | True | 2 | 5 |
| 74 | 356 | 1743 | datetime int_da... | datetime int_da... | True | 2 | 8 |
| 130 | 1464 | 1445 | datetime int_da... | datetime int_dat... | True | 2 | 16 |
| ... | ... | ... | ... | ... | ... | ... | ... |
| 1391 | 154 | 512 | datetime int_dat... | datetime int_datet... | True | 2 | 6 |
| 1444 | 521 | 505 | datetime int_dat... | datetime int_dat... | True | 6 | 16 |
| 1445 | 665 | 505 | datetime int_dat... | datetime int_dat... | True | 3 | 28 |
| 1455 | 1227 | 502 | datetime int_dat... | datetime int_datet... | True | 2 | 44 |
| 1463 | 503 | 501 | datetime int_dat... | datetime int_dat... | True | 2 | 94 |

78 rows × 7 columns

# ДОДАТОК В

## Відомість кваліфікаційної роботи

| Позначення | Найменування | Дод. відомості |
|---|---|---|
| | Текстові документи | |
| **1.** | Пояснювальна записка | с. |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | Інші документи | |
| | | |
| **2.** | Презентаційні матеріали | плакатів |
| | | |
| | | |
| | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | | | | |
| | | | | | | | |
| . | . | Прізвище та ініціали. | Підп. | Дата | | | |
| Розробив | | Сушко А.О. | | | Інтелектуальний парсер для перетворення макетів web-сторінок у HTML розмітку | Шифр групи | Код напр./спец. |
| Перевірив | | Шевченко О.Ю. | | | | СШІм-20-3 | 122 |
| Н.контр. | | Малєєва І.А. | | | | ХНУРЕ кафедра ШІ | |
| Затв. | | Філатов В.О. | | | | | |