

Untitled5

October 13, 2022

```
[2]: import glob
files = glob.glob("exported-flows/*.csv")
print(files)
```

```
['exported-flows/ft-v05.2016-05-26.210712+0200.csv', 'exported-
flows/ft-v05.2016-05-26.161912+0200.csv', 'exported-
flows/ft-v05.2016-05-26.230225+0200.csv', 'exported-
flows/ft-v05.2016-05-26.200936+0200.csv', 'exported-
flows/ft-v05.2016-05-26.191201+0200.csv', 'exported-
flows/ft-v05.2016-05-26.152136+0200.csv', 'exported-
flows/ft-v05.2016-05-26.220448+0200.csv', 'exported-
flows/ft-v05.2016-05-26.181424+0200.csv', 'exported-
flows/ft-v05.2016-05-26.171648+0200.csv', 'exported-
flows/ft-v05.2016-05-26.142400+0200.csv']
```

```
[18]: total_amount_of_entries = 0
total_amount_of_packets = 0
source_ips = set([])
dest_ips = set([])
list_of_protocols = []
list_of_ports = []
port_dest_ips = dict()

import networkx as nx
G = nx.Graph()

import csv
for exported_flow in files:
    with open(exported_flow, "r") as f:
        data = csv.reader(f, delimiter=",")
        next(data)
        for line in data:
            total_amount_of_entries += 1
            total_amount_of_packets += int(line[4])

            source_ip = line[10]
            dest_ip = line[11]
            if source_ip not in source_ips:
```

```

        source_ips.add(source_ip)
    if dest_ip not in dest_ips:
        dest_ips.add(dest_ip)

    protocol = int(line[17])
    list_of_protocols.append(protocol)

    if protocol == 6 or protocol == 17:
        G.add_edge(source_ip, dest_ip)
        src_port = int(line[15])
        dest_port = int(line[16])
        list_of_ports.append(src_port)
        list_of_ports.append(dest_port)
        if dest_port in port_dest_ips:
            port_dest_ips[dest_port].append(dest_ip)
        else:
            port_dest_ips[dest_port] = [dest_ip]

print(total_amount_of_entries)
print(total_amount_of_packets)
print(len(source_ips))

```

```

1073961
33345756
7918

```

```

[19]: import seaborn as sns
import matplotlib.pyplot as plt
plt.xlim((0, 1024))
plt.ylim(0, 1000)
sns.distplot(list_of_protocols, kde=False)

```

```

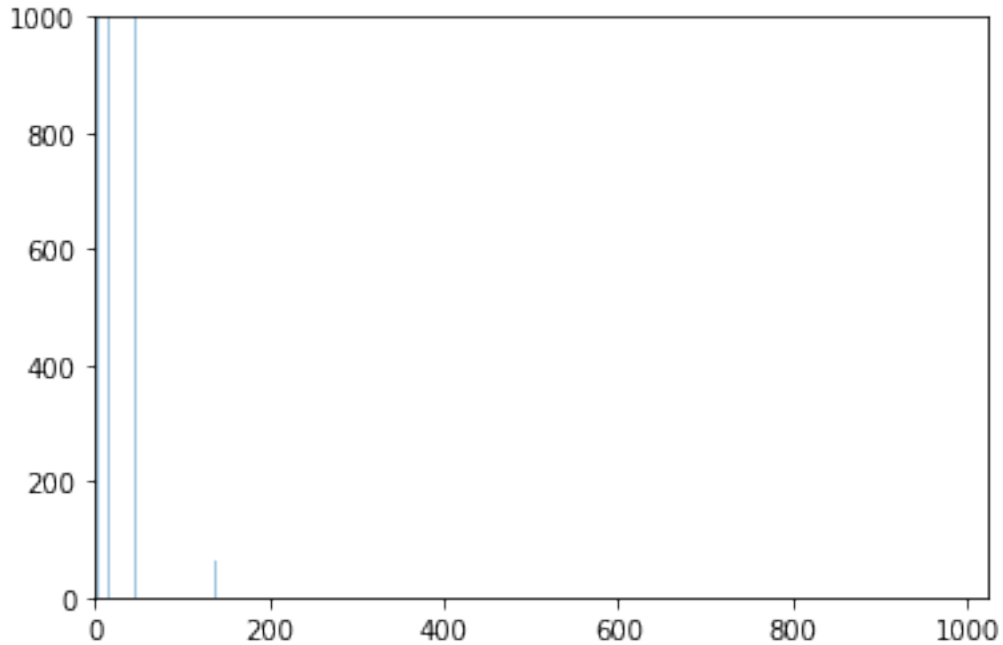
/home/matej/.local/lib/python3.8/site-packages/seaborn/distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)

```

```

[19]: <AxesSubplot:>

```



```
[20]: from collections import Counter
a = Counter(list_of_ports)
a.most_common(10)
a[54]
```

[20]: 7

```
[21]: # filtrovani portu
filtered_ports = [port for port in list_of_ports if port < 1024]

filtered_ports = []
for port in list_of_ports:
    if port < 1024:
        filtered_ports.append(port)
```

```
[22]: tmp = set(port_dest_ips[54])
len(tmp)
```

[22]: 2

```
[24]: for port in set(filtered_ports):
    if port in port_dest_ips:
        tmp = set(port_dest_ips[port])
        if (len(tmp) > 10 and a[port] > 200):
            # do (6)
```

```
print(port)
```

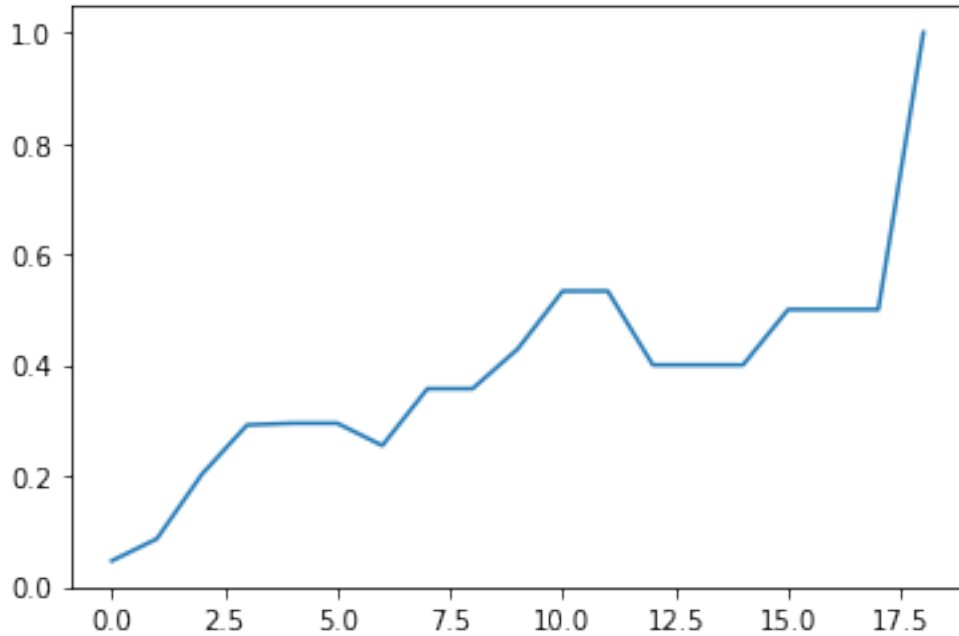
```
53  
80  
123  
137  
138  
161  
443  
445  
993
```

```
[29]: selected_ports = []  
counter_of_ports = Counter(list_of_ports)  
for port in counter_of_ports.most_common():  
    if port[1] <= 200:  
        continue  
    if port[0] > 1024:  
        continue  
    list_ip_addresses = port_dest_ips[port[0]]  
    num_of_unique = len(set(list_ip_addresses))  
    if num_of_unique > 10:  
        print(port[0], " : ", num_of_unique)  
        selected_ports.append(port[0])
```

```
53 : 53  
443 : 2383  
445 : 15  
80 : 2097  
123 : 79  
161 : 30  
137 : 37  
993 : 36  
138 : 14
```

```
[41]: my_port = 123  
all_my_port_ips = set(port_dest_ips[my_port])  
my_subgraph = G.subgraph(list(all_my_port_ips))  
#nx.draw(my_subgraph, node_size=100)  
rich_club = nx.rich_club_coefficient(my_subgraph, normalized=False)  
x_values = rich_club.keys()  
y_values = rich_club.values()  
import matplotlib.pyplot as plt  
plt.plot(x_values, y_values)
```

```
[41]: [<matplotlib.lines.Line2D at 0x7f7ad4d3efa0>]
```



```
[42]: rich_club
```

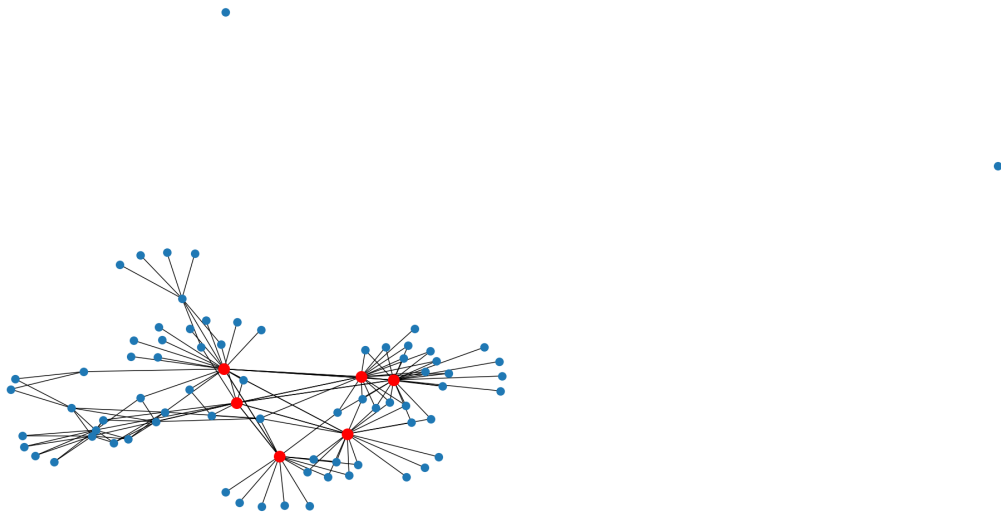
```
[42]: {0: 0.04647983595352016,
1: 0.08627450980392157,
2: 0.2028985507246377,
3: 0.2916666666666667,
4: 0.2948717948717949,
5: 0.2948717948717949,
6: 0.2545454545454545,
7: 0.35714285714285715,
8: 0.35714285714285715,
9: 0.42857142857142855,
10: 0.5333333333333333,
11: 0.5333333333333333,
12: 0.4,
13: 0.4,
14: 0.4,
15: 0.5,
16: 0.5,
17: 0.5,
18: 1.0}
```

```
[36]: rich_club_nodes = []
for node in my_subgraph.nodes():
    if my_subgraph.degree[node] >= 12:
        rich_club_nodes.append(node)
```

```
print(rich_club_nodes)
```

```
['195.113.144.201', '224.0.1.1', '147.32.211.2', '147.32.211.33',  
'147.32.211.24', '147.32.211.18']
```

```
[39]: positioning = nx.spring_layout(my_subgraph)  
plt.figure(figsize=(20, 10), dpi=100)  
nx.draw(my_subgraph, pos=positioning, node_size=100)  
club = G.subgraph(rich_club_nodes)  
nx.draw(club, pos=positioning, node_size=200, node_color="red")
```



```
[ ]:
```