research_brief

September 27, 2020

1 Training of dataset from Kaggle

[1]:
```
import pandas as pd
import numpy as np
data = pd.read_csv("final_dataset.csv")
data.head()
```

[1]:

	Unnamed: 0	Flow ID	Src IP	Src Port	\
0	624	192.168.4.118-203.73.24.75-4504-80-6	192.168.4.118	4504	
1	625	192.168.4.118-203.73.24.75-4504-80-6	192.168.4.118	4504	
2	626	192.168.4.118-203.73.24.75-4505-80-6	192.168.4.118	4505	
3	627	192.168.4.118-203.73.24.75-4505-80-6	192.168.4.118	4505	
4	628	192.168.4.118-203.73.24.75-4506-80-6	192.168.4.118	4506	

	Dst IP	Dst Port	Protocol	Timestamp	Flow Duration	\
0	203.73.24.75	80	6	12/06/2010 08:34:32 AM	3974862	
1	203.73.24.75	80	6	12/06/2010 08:34:36 AM	63	
2	203.73.24.75	80	6	12/06/2010 08:34:36 AM	476078	
3	203.73.24.75	80	6	12/06/2010 08:34:37 AM	151	
4	203.73.24.75	80	6	12/06/2010 08:34:37 AM	472507	

	Tot Fwd Pkts	…	Fwd Seg Size Min	Active Mean	Active Std	Active Max	\
0	29	…	0	0.0	0.0	0.0	
1	1	…	0	0.0	0.0	0.0	
2	2	…	0	0.0	0.0	0.0	
3	2	…	0	0.0	0.0	0.0	
4	2	…	0	0.0	0.0	0.0	

	Active Min	Idle Mean	Idle Std	Idle Max	Idle Min	Label
0	0.0	0.0	0.0	0.0	0.0	ddos
1	0.0	0.0	0.0	0.0	0.0	ddos
2	0.0	0.0	0.0	0.0	0.0	ddos
3	0.0	0.0	0.0	0.0	0.0	ddos
4	0.0	0.0	0.0	0.0	0.0	ddos

[5 rows x 85 columns]

2 Column Names of dataset

[2]:
```
for sample in data:
    print(sample)
```

```
Unnamed: 0
Flow ID
Src IP
Src Port
Dst IP
Dst Port
Protocol
Timestamp
Flow Duration
Tot Fwd Pkts
Tot Bwd Pkts
TotLen Fwd Pkts
TotLen Bwd Pkts
Fwd Pkt Len Max
Fwd Pkt Len Min
Fwd Pkt Len Mean
Fwd Pkt Len Std
Bwd Pkt Len Max
Bwd Pkt Len Min
Bwd Pkt Len Mean
Bwd Pkt Len Std
Flow Byts/s
Flow Pkts/s
Flow IAT Mean
Flow IAT Std
Flow IAT Max
Flow IAT Min
Fwd IAT Tot
Fwd IAT Mean
Fwd IAT Std
Fwd IAT Max
Fwd IAT Min
Bwd IAT Tot
Bwd IAT Mean
Bwd IAT Std
Bwd IAT Max
Bwd IAT Min
Fwd PSH Flags
Bwd PSH Flags
Fwd URG Flags
Bwd URG Flags
Fwd Header Len
Bwd Header Len
```

Fwd Pkts/s
Bwd Pkts/s
Pkt Len Min
Pkt Len Max
Pkt Len Mean
Pkt Len Std
Pkt Len Var
FIN Flag Cnt
SYN Flag Cnt
RST Flag Cnt
PSH Flag Cnt
ACK Flag Cnt
URG Flag Cnt
CWE Flag Count
ECE Flag Cnt
Down/Up Ratio
Pkt Size Avg
Fwd Seg Size Avg
Bwd Seg Size Avg
Fwd Byts/b Avg
Fwd Pkts/b Avg
Fwd Blk Rate Avg
Bwd Byts/b Avg
Bwd Pkts/b Avg
Bwd Blk Rate Avg
Subflow Fwd Pkts
Subflow Fwd Byts
Subflow Bwd Pkts
Subflow Bwd Byts
Init Fwd Win Byts
Init Bwd Win Byts
Fwd Act Data Pkts
Fwd Seg Size Min
Active Mean
Active Std
Active Max
Active Min
Idle Mean
Idle Std
Idle Max
Idle Min
Label

3 Dimension of Total Dataset , Normal Data and Attack Data

[3]:
```
l = data.shape
print("Total Dataset:")
print("Number of rows is ",l[0]," and columns is ",l[1])
data_normal = data[data["Label"]=="Benign"]
data_ddos = data[data["Label"]=="ddos"]
l1 = data_normal.shape
print("Good Dataset:")
print("Number of rows is ",l1[0]," and columns is ",l1[1])
l2 = data_ddos.shape
print("Bad Dataset:")
print("Number of rows is ",l2[0]," and columns is ",l2[1])
```

```
Total Dataset:
Number of rows is  12794627  and columns is  85
Good Dataset:
Number of rows is  6321980  and columns is  85
Bad Dataset:
Number of rows is  6472647  and columns is  85
```

4 Attackers Source IP with their frequency in attack data

[4]:
```
m = np.array(data_ddos['Src IP'].unique())
n = len(m)
count_ddos = [0]*n
count = 0
for i in range(0,n):
    count_ddos[i] =  data_ddos['Src IP'].isin([m[i]]).sum()
print("Attackers Dataset:")
for i in range (0,n):
    print(m[i], " "*10,count_ddos[i])
for i in range(0,n):
    count = count + count_ddos[i]
print("Total DDOS attacks are: ", count)
```

```
Attackers Dataset:
192.168.4.118            149
192.168.1.103            744
192.168.2.108            1956
192.168.2.110            5924
192.168.2.112            153
192.168.4.121            12
192.168.1.101            1525
192.168.1.104            2559
192.168.56.102            4990
192.168.56.1            19703
```

```
192.168.3.114            2650
192.168.3.115            45
192.168.2.109            9407
192.168.4.119            1666
192.168.4.120            655
192.168.3.117            920
192.168.1.105            86
192.168.1.102            360
192.168.2.113            24
192.168.2.111            28
18.219.211.138            41508
18.217.165.70            10990
172.31.69.25            1766461
18.219.5.43             181432
18.218.55.126            182462
18.216.200.189            183850
52.14.136.135            182177
18.219.9.1              183140
18.218.11.51             182325
18.216.24.42             182256
18.219.32.43             181729
18.218.115.60            180500
18.218.229.235            184084
172.31.69.28            920151
13.59.126.31             105550
18.219.193.20            1750476
Total DDOS attacks are:  6472647
```

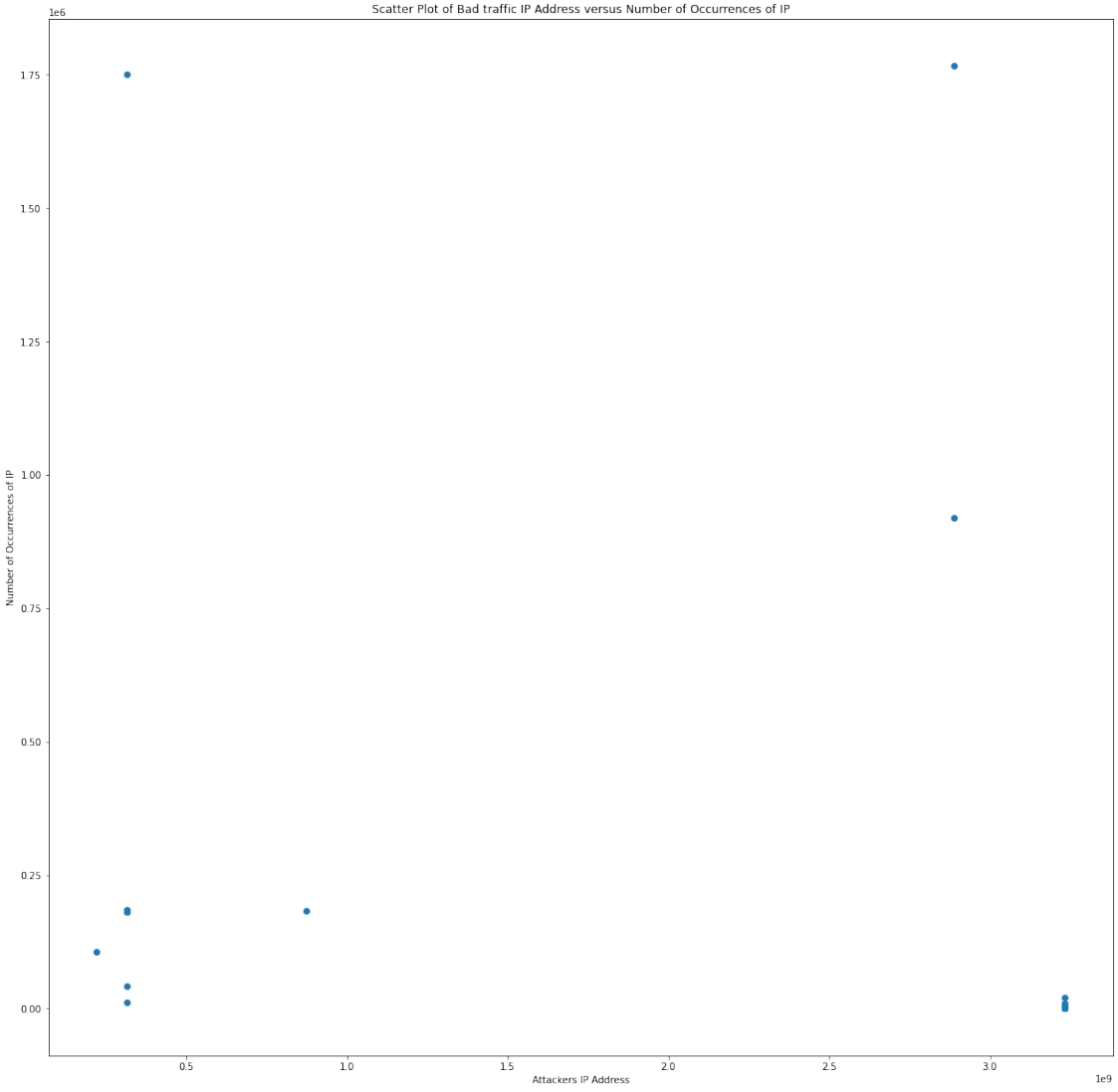
[5]:
```
import socket
import struct


def ip2int(addr):
    return struct.unpack("!I", socket.inet_aton(addr))[0]


def int2ip(addr):
    return socket.inet_ntoa(struct.pack("!I", addr))

x = [0]*n
y = [0]*n
for i in range(0,n):
    x[i] = ip2int(m[i])
    y[i] = count_ddos[i]
print("-----------------------------------------------")
print( "Integer IP address   Number of Occurrences")
for i in range (0,n):
```

```
    print(x[i]," "*10,y[i])
print("-----------------------------------------------")
```

```
-----------------------------------------------
Integer IP address   Number of Occurrences
3232236662            149
3232235879            744
3232236140            1956
3232236142            5924
3232236144            153
3232236665            12
3232235877            1525
3232235880            2559
3232249958            4990
3232249857            19703
3232236402            2650
3232236403            45
3232236141            9407
3232236663            1666
3232236664            655
3232236405            920
3232235881            86
3232235878            360
3232236145            24
3232236143            28
316396426            41508
316253510            10990
2887730457            1766461
316343595            181432
316290942            182462
316197053            183850
873367687            182177
316344577            183140
316279603            182325
316151850            182256
316350507            181729
316306236            180500
316335595            184084
2887730460            920151
222002719            105550
316391700            1750476
-----------------------------------------------
```

[6]:
```
import numpy as np
import matplotlib.pyplot as plt
plt.figure(figsize=(20,20))
plt.scatter(x,y)
```

```
plt.title("Scatter Plot of Bad traffic IP Address versus Number of Occurrences␣
 ↪of IP")
plt.xlabel("Attackers IP Address")
plt.ylabel("Number of Occurrences of IP")
plt.show()
```



[20]:
```
p = [0]*20
q = [0]*20
r = [0]*13
s = [0]*13
count = 0
count1 = 0
print("-----------------------------------------------------")
```

```
print("   IP address                    Number of Occurence   ")
for i in range(0,n):
    if(x[i]>3*10**9 and y[i] <0.25*(10**6)):
        p[count] = x[i]
        q[count] = y[i]
        count = count + 1
        print(int2ip(x[i])," "*20, y[i])
    if(x[i]<10**9 and y[i] <0.25*(10**6)):
        r[count1] = x[i]
        s[count1] = y[i]
        count1 = count1 + 1
        print(int2ip(x[i])," "*20, y[i])
print("-----------------------------------------------------------")
print("Total Count of attackers having higher chance of occuring in the attack␣
 ↪is",count+count1)
```

```
-----------------------------------------------------------
   IP address                    Number of Occurence
192.168.4.118                      149
192.168.1.103                      744
192.168.2.108                      1956
192.168.2.110                      5924
192.168.2.112                      153
192.168.4.121                      12
192.168.1.101                      1525
192.168.1.104                      2559
192.168.56.102                      4990
192.168.56.1                      19703
192.168.3.114                      2650
192.168.3.115                      45
192.168.2.109                      9407
192.168.4.119                      1666
192.168.4.120                      655
192.168.3.117                      920
192.168.1.105                      86
192.168.1.102                      360
192.168.2.113                      24
192.168.2.111                      28
18.219.211.138                      41508
18.217.165.70                      10990
18.219.5.43                     181432
18.218.55.126                      182462
18.216.200.189                      183850
52.14.136.135                      182177
18.219.9.1                     183140
18.218.11.51                      182325
18.216.24.42                      182256
```

```
18.219.32.43                        181729
18.218.115.60                        180500
18.218.229.235                        184084
13.59.126.31                        105550
------------------------------------------------------------
Total Count of attackers having higher chance of occuring in the attack is 33
```

5 Overview on attackers IP address range with maximum probability occurred in bad traffic recognized real-time

[23]:
```
maxip = max(p)
minip = min(p)
maxiq = max(q)
miniq = min(q)
maxip1 = max(r)
minip1 = min(r)
maxiq1 = max(s)
miniq1 = min(s)
print("The probability of Attackers IP address in the range from ",␣
 ↪int2ip(minip1), " to ", int2ip(maxip1), " and from ", int2ip(minip), " to ",␣
 ↪int2ip(maxip)," is ", 33/36*100 ,"%")
```

```
The probability of Attackers IP address in the range from  13.59.126.31  to
52.14.136.135  and from  192.168.1.101  to  192.168.56.102  is
91.66666666666666 %
```