UDP_plaintext_RFC761

September 28, 2021

[ ]:
```
import pandas as pd
from numpy import arange
import matplotlib.pyplot as plt
%matplotlib inline
```

[ ]:
```
# Read from CSV

df = pd.read_csv('Outputs/packets-udp-rfc761-plaintext-dinamico-2021.09.
↪27-processed')
```

[ ]:
```
# Setting global var

bytesize = 32
proto = 'UDP'
dstport = 20001
encoding = 'plaintext'
text = 'RFC761'
```

[ ]:
```
# Add a new column to the end called 'flow'

df['flow'] = df['srcip'] + ':' + df.srcport.map(str) + ' -> ' + df['dstip'] + ':
↪' + df.dstport.map(str)
# Read a specific location (R,C)
print('Example of flow {}'.format(df.iloc[5,10]))
```

```
Example of flow 127.0.0.1:61132 -> 127.0.0.1:20001
```

[ ]:
```
# Sort dataframe by an index (column) and show

df = df.sort_values(['payload_size','flow'])
print(df.iloc[:,6:11])
```

```
     payload_size   shannon      bien     tbien  \
40              1  1.000000  0.468917  0.759649
41              1  1.000000  0.468917  0.759649
42              1  1.000000  0.468917  0.759649
43              1  1.000000  0.468917  0.759649
39            225  0.738205  0.117504  0.909993
..            …       …       …       …
```

```
156          1032  0.800705  0.234534  0.928275
157          1032  0.738205  0.926666  0.962281
158          1032  0.681804  0.926631  0.959906
159          1032  0.681804  0.459282  0.946817
160          1032  0.706955  0.477043  0.956688

                                    flow
40   127.0.0.1:49792 -> 127.0.0.1:49791
41   127.0.0.1:49792 -> 127.0.0.1:49791
42   127.0.0.1:49792 -> 127.0.0.1:49791
43   127.0.0.1:49792 -> 127.0.0.1:49791
39   127.0.0.1:61132 -> 127.0.0.1:20001
..                                   …
156  127.0.0.1:61132 -> 127.0.0.1:20001
157  127.0.0.1:61132 -> 127.0.0.1:20001
158  127.0.0.1:61132 -> 127.0.0.1:20001
159  127.0.0.1:61132 -> 127.0.0.1:20001
160  127.0.0.1:61132 -> 127.0.0.1:20001

[161 rows x 5 columns]
```

[ ]:
```
# Filtering by port

is_port = df['dstport']==dstport
print(is_port.head())
df = df[is_port]
```

```
40    False
41    False
42    False
43    False
39     True
Name: dstport, dtype: bool
```

[ ]:
```
# Filtering by the number of packets of chosen size

is_bytes = df['payload_size']>2
print(is_bytes.head())
df = df[is_bytes]
```

```
39    True
0     True
1     True
2     True
3     True
Name: payload_size, dtype: bool
```

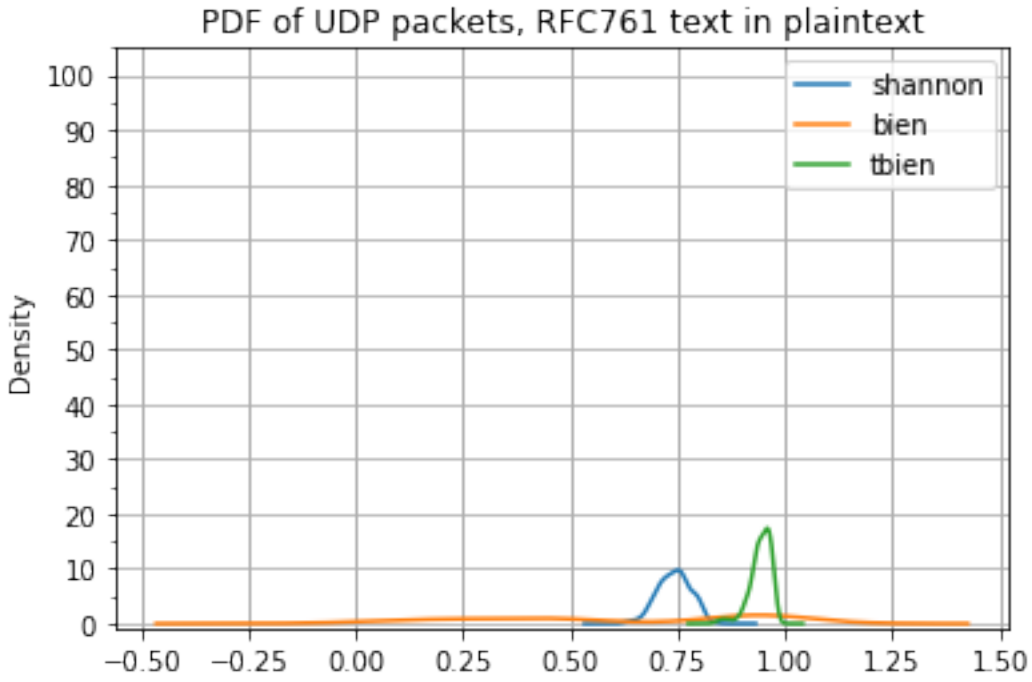
[ ]:
```
# Minimize number of displayed columns

# pd.set_option("display.max.columns", None)
# df.head()
```

[ ]:
```
# Aggregation by flow and each entropies mean

df[['flow','shannon','bien','tbien','payload_size']].groupby('flow').mean().
 ↪sort_values('tbien', ascending=False)
```

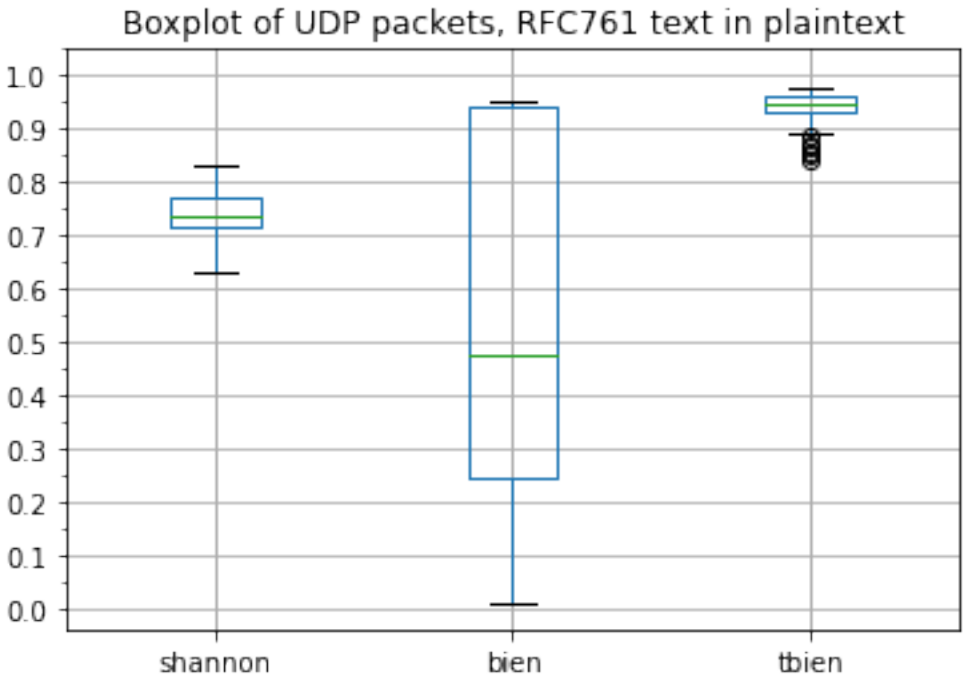
[ ]:

flow	shannon	bien	tbien	payload_size
127.0.0.1:61132 -> 127.0.0.1:20001	0.741862	0.609778	0.943431	1026.859873

[ ]:
```
# Plot 1

title = 'PDF of {} packets, {} text in {}'.format(proto, text, encoding)
ax = df.plot(x='payload_size',␣
 ↪y=['shannon','bien','tbien'],kind='density',title=title, grid=True)
ax.xaxis.grid(True, which='major', linestyle='-', linewidth=1)
ymajortick = arange(0,110,10)
yminortick = arange(0,110,5)
ax.set_yticks( ymajortick, minor=False )
ax.set_yticks( yminortick, minor=True )
ax.grid('on', which='both', axis='x' )
plt.savefig('Plots/rfc761/{}{}{}density.png'.format(proto, encoding, text),␣
 ↪transparent=False)
```



[ ]:
```
# Plot 2

title = 'Boxplot of {} packets, {} text in {}'.format(proto, text, encoding)
ax = df.plot(x='payload_size',␣
 ↪y=['shannon','bien','tbien'],kind='box',title=title, grid=True)
ax.xaxis.grid(True, which='major', linestyle='-', linewidth=1)
ymajortick = arange(0,1.1,0.1)
yminortick = arange(0,1.1,0.05)
ax.set_yticks( ymajortick, minor=False )
ax.set_yticks( yminortick, minor=True )
ax.grid('on', which='both', axis='x' )
plt.savefig('Plots/rfc761/{}{}{}box.png'.format(proto, encoding, text),␣
 ↪transparent=False)
```



[ ]:
```
# Table of data

df = df.describe()
print(df)
```

```
       srcport  dstport  payload_size     shannon        bien       tbien
count    157.0    157.0    157.000000  157.000000  157.000000  157.000000
mean   61132.0  20001.0   1026.859873    0.741862    0.609778    0.943431
```

```
std        0.0      0.0     64.405612    0.037184    0.332511    0.024937
min    61132.0  20001.0    225.000000    0.632660    0.007554    0.841906
25%    61132.0  20001.0   1032.000000    0.714615    0.245087    0.932449
50%    61132.0  20001.0   1032.000000    0.738205    0.472996    0.946168
75%    61132.0  20001.0   1032.000000    0.769455    0.942447    0.961526
max    61132.0  20001.0   1032.000000    0.831955    0.953435    0.975157
```

[ ]:
```
# Exporting new data

filename = 'Outputs/RFC761/{}{}{}data.csv'.format(proto, encoding, text)
df.to_csv(filename,',')
```