In [1]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:

```
dataset= pd.read_csv('E:\APA-DDoS-Dataset.csv')
```

In [3]:

```
dataset
```

Out[3]:

	ip.src	ip.dst	tcp.srcport	tcp.dstport	ip.proto	frame.len	tcp.flags.syn	tcp.flags.reset	tcp.flags.push	tc
0	192.168.1.1	192.168.23.2	2412	8000	6	54	0	0	1	
1	192.168.1.1	192.168.23.2	2413	8000	6	54	0	0	1	
2	192.168.1.1	192.168.23.2	2414	8000	6	54	0	0	1	
3	192.168.1.1	192.168.23.2	2415	8000	6	54	0	0	1	
4	192.168.1.1	192.168.23.2	2416	8000	6	54	0	0	1	
...	...	...	...	...	...	...	...	...	...	...
151195	192.168.19.1	192.168.23.2	37360	8000	6	66	0	0	0	
151196	192.168.19.1	192.168.23.2	37362	8000	6	66	0	0	0	
151197	192.168.19.1	192.168.23.2	37364	8000	6	66	0	0	0	
151198	192.168.19.1	192.168.23.2	37366	8000	6	66	0	0	0	
151199	192.168.19.1	192.168.23.2	37368	8000	6	66	0	0	0	

151200 rows × 23 columns

In [4]:

```
print(dataset.shape)
```

```
(151200, 23)
```

In [5]:

```
dataset.columns
```

```
dataset.columns
```

Out[5]:

```
Index(['ip.src', 'ip.dst', 'tcp.srcport', 'tcp.dstport', 'ip.proto',
       'frame.len', 'tcp.flags.syn', 'tcp.flags.reset', 'tcp.flags.push',
       'tcp.flags.ack', 'ip.flags.mf', 'ip.flags.df', 'ip.flags.rb', 'tcp.seq',
       'tcp.ack', 'frame.time', 'Packets', 'Bytes', 'Tx Packets', 'Tx Bytes',
       'Rx Packets', 'Rx Bytes', 'Label'],
      dtype='object')
```

In [6]:

```
## to capture any nan values in the dataset
features_nan=[feature for feature in dataset.columns if dataset[feature].isnull().sum()>
1 and dataset[feature].dtypes=='O']
if len(features_nan) !=0:
    for feature in features_nan:
        print("{} has: {}% missing values".format(feature,np.round(dataset[feature].isnu
ll().mean(),4)*100))
else :
    print("no such feature")
```

```
no such feature
```

In [7]:

```
numerical_features=[feature for feature in dataset.columns if dataset[feature].dtypes!='
O']
print("number of numerical variables ",len(numerical_features))
```

```
number of numerical variables  19
```

In [8]:

```
dataset[numerical_features].head()
```

Out[8]:

	tcp.srcport	tcp.dstport	ip.proto	frame.len	tcp.flags.syn	tcp.flags.reset	tcp.flags.push	tcp.flags.ack	ip.flags.mf	ip.flags.d
0	**2412**	**8000**	**6**	**54**	**0**	**0**	**1**	**1**	**0**	**0**
1	**2413**	**8000**	**6**	**54**	**0**	**0**	**1**	**1**	**0**	**0**
2	**2414**	**8000**	**6**	**54**	**0**	**0**	**1**	**1**	**0**	**0**
3	**2415**	**8000**	**6**	**54**	**0**	**0**	**1**	**1**	**0**	**0**
4	**2416**	**8000**	**6**	**54**	**0**	**0**	**1**	**1**	**0**	**0**

In [9]:

```
#values of some features has same value throughout the dataset and does not affect labels
,so dropping them
dataset=dataset.drop(columns=["tcp.dstport","ip.proto","tcp.flags.syn","tcp.flags.reset"
,"tcp.flags.ack","ip.flags.mf","ip.flags.rb","tcp.seq","tcp.ack","frame.time"])
```

In [10]:

```
dataset.columns
```

Out[10]:

```
Index(['ip.src', 'ip.dst', 'tcp.srcport', 'frame.len', 'tcp.flags.push',
       'ip.flags.df', 'Packets', 'Bytes', 'Tx Packets', 'Tx Bytes',
       'Rx Packets', 'Rx Bytes', 'Label'],
      dtype='object')
```

In [11]:

```
dataset.tail()
```

Out[11]:

	ip.src	ip.dst	tcp.srcport	frame.len	tcp.flags.push	ip.flags.df	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets
151195	192.168.19.1	192.168.23.2	37360	66	0	1	10	1146	6	560	4
151196	192.168.19.1	192.168.23.2	37362	66	0	1	10	1151	6	560	4
151197	192.168.19.1	192.168.23.2	37364	66	0	1	10	1144	6	560	4
151198	192.168.19.1	192.168.23.2	37366	66	0	1	10	1175	6	560	4
151199	192.168.19.1	192.168.23.2	37368	66	0	1	10	1146	6	560	4

In [12]:

```
label_dummy=pd.get_dummies(dataset['Label'])
```

In [13]:

```
dataset=pd.concat([dataset,label_dummy], axis=1)
```

In [14]:

```
from sklearn.preprocessing import LabelEncoder
```

In [15]:

```
le=LabelEncoder()
```

In [16]:

```
dataset['DDoS-PSH-ACK']=le.fit_transform(dataset['DDoS-PSH-ACK'])
dataset['Benign']=le.fit_transform(dataset['Benign'])
dataset['DDoS-ACK']=le.fit_transform(dataset['DDoS-ACK'])
```

In [17]:

```
dataset.tail()
```

Out[17]:

	ip.src	ip.dst	tcp.srcport	frame.len	tcp.flags.push	ip.flags.df	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets
151195	192.168.19.1	192.168.23.2	37360	66	0	1	10	1146	6	560	4
151196	192.168.19.1	192.168.23.2	37362	66	0	1	10	1151	6	560	4
151197	192.168.19.1	192.168.23.2	37364	66	0	1	10	1144	6	560	4
151198	192.168.19.1	192.168.23.2	37366	66	0	1	10	1175	6	560	4
151199	192.168.19.1	192.168.23.2	37368	66	0	1	10	1146	6	560	4

In [18]:

```
feature_scale=[feature for feature in dataset.columns if feature not in ['ip.src','ip.ds
t','Benign','DDoS-ACK','DDoS-PSH-ACK','Label']]
```

In [19]:

```
from sklearn.preprocessing import StandardScaler
```

In [20]:

```
scaler=StandardScaler()
```

In [21]:

```
scaler.fit(dataset[feature_scale])
```

Out[21]:

▾ StandardScaler
StandardScaler()

In [22]:

```
scaler.transform(dataset[feature_scale])
```

Out[22]:

```
array([[-1.27146315, -0.63214064,  1.        , ..., -0.98567572,
        -0.03518717, -0.97784837],
       [-1.27141222, -0.63214064,  1.        , ..., -0.68097412,
         1.00312277, -0.70388293],
       [-1.27136129, -0.63214064,  1.        , ..., -0.37627252,
         2.0414327 , -0.4299175 ],
       ...,
       [ 0.50864024, -0.46366387, -1.        , ...,  0.95538634,
        -0.03518717,  0.88917535],
       [ 0.5087421 , -0.46366387, -1.        , ...,  0.95538634,
        -0.03518717,  1.0464518 ],
       [ 0.50884396, -0.46366387, -1.        , ...,  0.95538634,
        -0.03518717,  0.89932222]])
```

In [23]:

```
dataset.head()
```

Out[23]:

	ip.src	ip.dst	tcp.srcport	frame.len	tcp.flags.push	ip.flags.df	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes
0	192.168.1.1	192.168.23.2	2412	54	1	0	8	432	4	216	4	216
1	192.168.1.1	192.168.23.2	2413	54	1	0	10	540	5	270	5	270
2	192.168.1.1	192.168.23.2	2414	54	1	0	12	648	6	324	6	324
3	192.168.1.1	192.168.23.2	2415	54	1	0	10	540	5	270	5	270
4	192.168.1.1	192.168.23.2	2416	54	1	0	6	324	3	162	3	162

In [24]:

```
dataset=pd.concat([dataset[['ip.src','ip.dst','Benign','DDoS-ACK','DDoS-PSH-ACK','Label'
]].reset_index(drop=True),pd.DataFrame(scaler.transform(dataset[feature_scale]),columns=
feature_scale)],
                  axis=1)
```

In [25]:

```
dataset.head()
```

Out[25]:

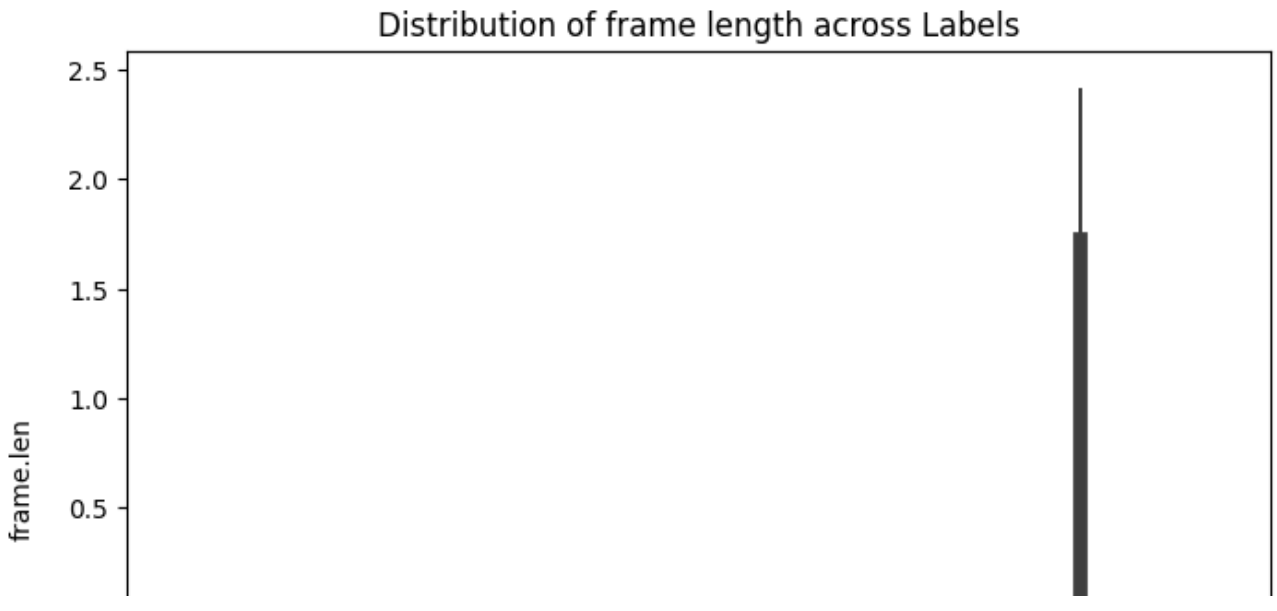
	ip.src	ip.dst	Benign	DDoS-ACK	DDoS-PSH-ACK	Label	tcp.srcport	frame.len	tcp.flags.push	ip.flags.df	Packets	By
0	192.168.1.1	192.168.23.2	0	0	1	DDoS-PSH-ACK	-1.271463	-0.632141	1.0	-1.0	-0.508386	0.983
1	192.168.1.1	192.168.23.2	0	0	1	DDoS-PSH-ACK	-1.271412	-0.632141	1.0	-1.0	0.430752	0.694
2	192.168.1.1	192.168.23.2	0	0	1	DDoS-PSH-ACK	-1.271361	-0.632141	1.0	-1.0	1.369890	0.405
3	192.168.1.1	192.168.23.2	0	0	1	DDoS-PSH-ACK	-1.271310	-0.632141	1.0	-1.0	0.430752	0.694
4	192.168.1.1	192.168.23.2	0	0	1	DDoS-PSH-ACK	-1.271259	-0.632141	1.0	-1.0	-1.447524	1.272

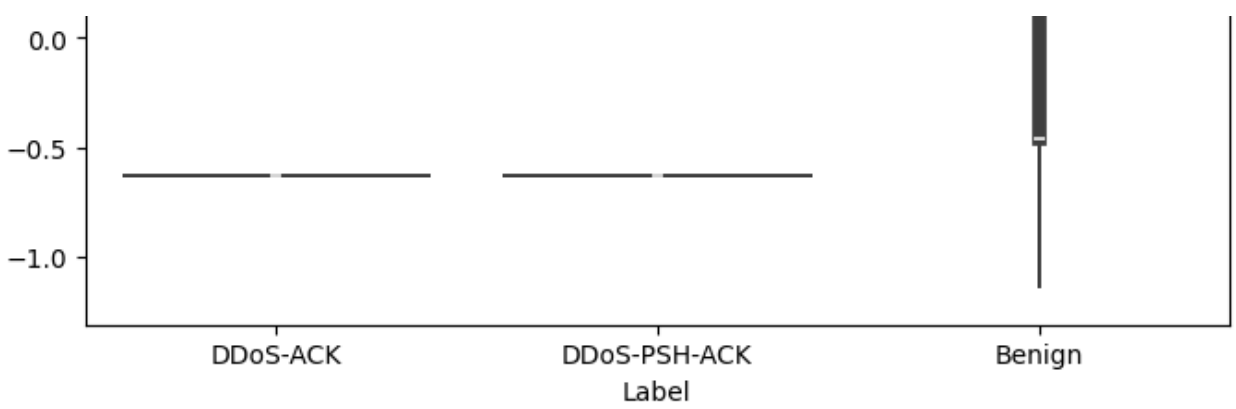
In [26]:

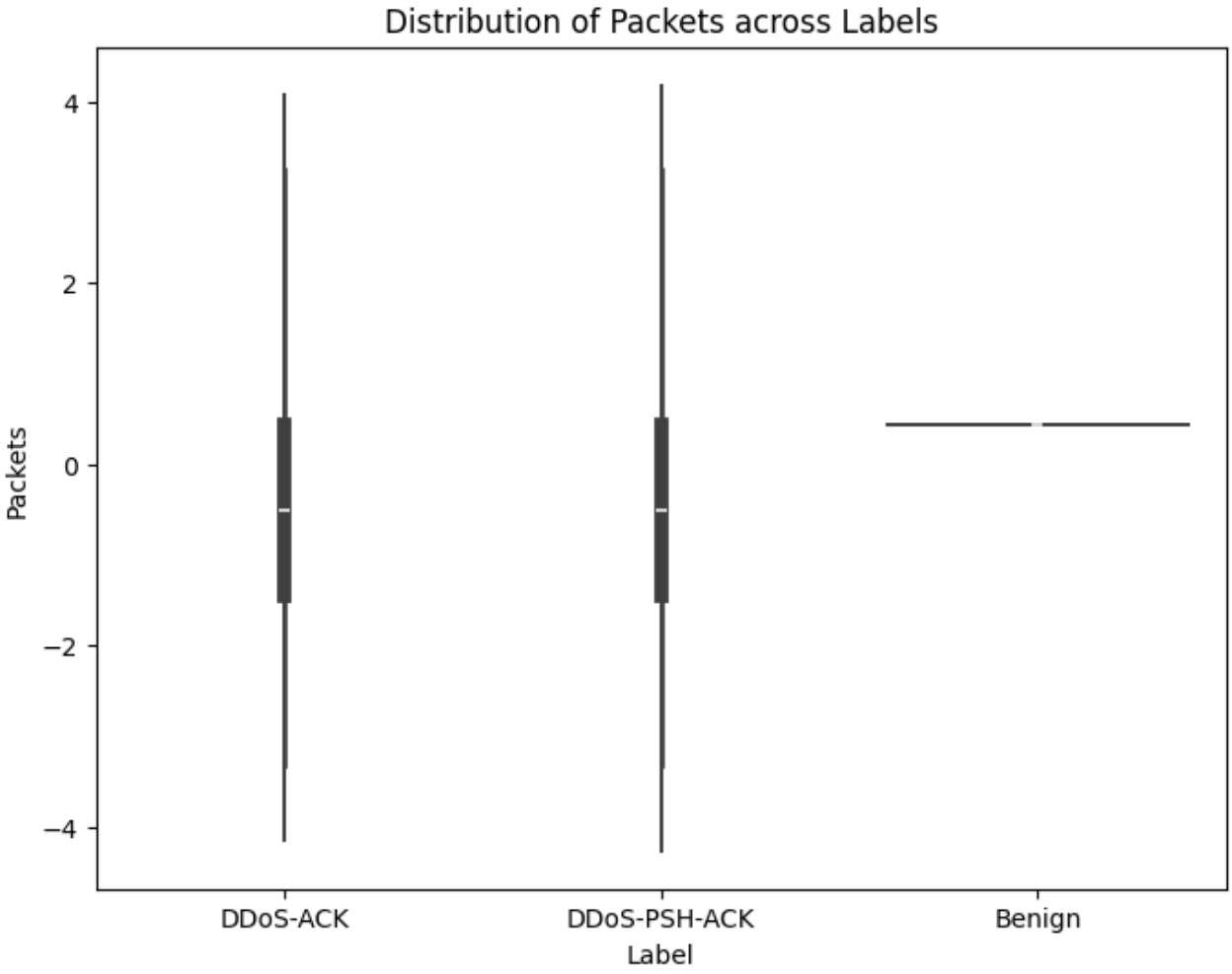
```
from sklearn.model_selection import train_test_split
rest_data, sampled_data = train_test_split(dataset, test_size=0.005, stratify=dataset['Label'], random_state=42)
```

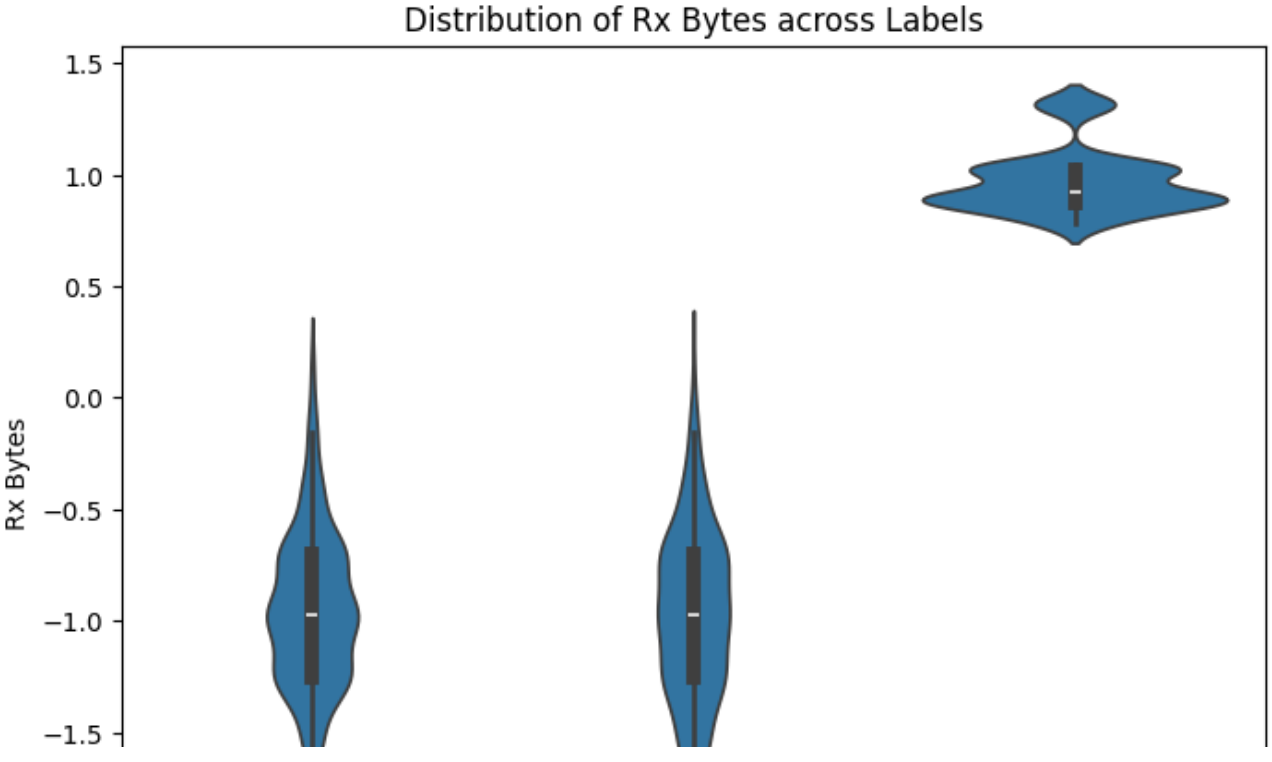
In [27]:

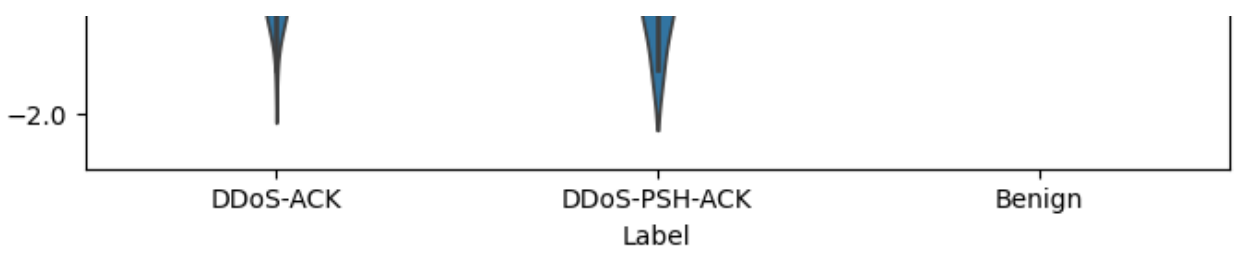
```
plt.figure(figsize=(8, 6))
sns.violinplot(data= sampled_data, x='Label', y='frame.len')
plt.title('Distribution of frame length across Labels')
plt.xlabel('Label')
plt.ylabel('frame.len')
plt.show()

plt.figure(figsize=(8, 6))
sns.violinplot(data= sampled_data, x='Label', y='Packets')
plt.title('Distribution of Packets across Labels')
plt.xlabel('Label')
plt.ylabel('Packets')
plt.show()

plt.figure(figsize=(8, 6))
sns.violinplot(data= sampled_data,  x='Label', y='Rx Bytes')
plt.title('Distribution of Rx Bytes across Labels')
plt.xlabel('Label')
plt.ylabel('Rx Bytes')
plt.show()
```




0.0
−0.5
−1.0
DDoS-ACK
DDoS-PSH-ACK
Benign
Label


Distribution of Packets across Labels
Packets
4
2
0
−2
−4
DDoS-ACK
DDoS-PSH-ACK
Benign
Label


Distribution of Rx Bytes across Labels
Rx Bytes
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5



In [28]:

```
dataset
```

Out[28]:

	ip.src	ip.dst	Benign	DDoS-ACK	DDoS-PSH-ACK	Label	tcp.srcport	frame.len	tcp.flags.push	ip.flags.df	Packets
0	192.168.1.1	192.168.23.2	0	0	1	DDoS-PSH-ACK	-1.271463	-0.632141	1.0	-1.0	-0.508386
1	192.168.1.1	192.168.23.2	0	0	1	DDoS-PSH-ACK	-1.271412	-0.632141	1.0	-1.0	0.430752
2	192.168.1.1	192.168.23.2	0	0	1	DDoS-PSH-ACK	-1.271361	-0.632141	1.0	-1.0	1.369890
3	192.168.1.1	192.168.23.2	0	0	1	DDoS-PSH-ACK	-1.271310	-0.632141	1.0	-1.0	0.430752
4	192.168.1.1	192.168.23.2	0	0	1	DDoS-PSH-ACK	-1.271259	-0.632141	1.0	-1.0	-1.447524
...	...	...	...	...	...	...	...	...	...	...	...
151195	192.168.19.1	192.168.23.2	1	0	0	Benign	0.508437	-0.463664	-1.0	1.0	0.430752
151196	192.168.19.1	192.168.23.2	1	0	0	Benign	0.508538	-0.463664	-1.0	1.0	0.430752
151197	192.168.19.1	192.168.23.2	1	0	0	Benign	0.508640	-0.463664	-1.0	1.0	0.430752
151198	192.168.19.1	192.168.23.2	1	0	0	Benign	0.508742	-0.463664	-1.0	1.0	0.430752
151199	192.168.19.1	192.168.23.2	1	0	0	Benign	0.508844	-0.463664	-1.0	1.0	0.430752

151200 rows × 16 columns

In [ ]:

```

```