Filename: _0_preprocessing.ipynb

Title: Intrusion Detection Prediction - Preprocessing

Author: Raghava | GitHub: @raghavtwenty

Date Created: June 10, 2023 | Last Updated: May 13, 2024

Language: Python | Version: 3.10.14, 64-bit

Importing Required Libraries

In [2]:

```python
import pandas as pd
import seaborn as sbn
import numpy as np
from sklearn.preprocessing import MinMaxScaler
from matplotlib import pyplot as mpl
```

Importing Dataset

In [3]:

```python
file_location = pd.read_csv("../datasets/raw_dataset.csv")

data_frame = pd.DataFrame(file_location)
```

View the dataset

In [4]:

```python
data_frame.head()
```

Out[4]:

	Port Number	Received Packets	Received Bytes	Sent Bytes	Sent Packets	Port alive Duration (S)	Packets Rx Dropped	Packets Tx Dropped
0	4	305111	25506841	100234870	284579	1657	0	0
1	2	209	20671	6316631	274	96	0	0
2	4	150	19774	6475473	3054	166	0	0
3	1	4699	100986365	124574097	413351	2267	0	0
4	3	990	104058	88896	778	792	0	0

5 rows × 32 columns

Know the detailed information about the dataset

In [5]:

```python
data_frame.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4927 entries, 0 to 4926
Data columns (total 32 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   Port Number                   4927 non-null   int64
 1   Received Packets              4927 non-null   int64
 2   Received Bytes                4927 non-null   int64
 3   Sent Bytes                    4927 non-null   int64
 4   Sent Packets                  4927 non-null   int64
 5   Port alive Duration (S)       4927 non-null   int64
 6   Packets Rx Dropped            4927 non-null   int64
 7   Packets Tx Dropped            4927 non-null   int64
 8   Packets Rx Errors             4927 non-null   int64
 9   Packets Tx Errors             4927 non-null   int64
 10  Delta Received Packets        4927 non-null   int64
 11  Delta Received Bytes          4927 non-null   int64
 12  Delta Sent Bytes              4927 non-null   int64
 13  Delta Sent Packets            4927 non-null   int64
 14  Delta Port alive Duration (S) 4927 non-null   int64
 15  Delta Packets Rx Dropped      4927 non-null   int64
 16  Delta Packets Tx Dropped      4927 non-null   int64
 17  Delta Packets Rx Errors       4927 non-null   int64
 18  Delta Packets Tx Errors       4927 non-null   int64
 19  Connection Point              4927 non-null   int64
 20  Total Load/Rate               4927 non-null   int64
 21  Total Load/Latest             4927 non-null   int64
 22  Unknown Load/Rate             4927 non-null   int64
 23  Unknown Load/Latest           4927 non-null   int64
 24  Latest bytes counter          4927 non-null   int64
 25  is_valid                      4927 non-null   int64
 26  Table ID                      4927 non-null   int64
 27  Active Flow Entries           4927 non-null   int64
 28  Packets Looked Up             4927 non-null   int64
 29  Packets Matched               4927 non-null   int64
 30  Max Size                      4927 non-null   int64
 31  Label                         4927 non-null   int64
dtypes: int64(32)
memory usage: 1.2 MB
```

Check for null values and corresponding count

In [6]:

```
data_frame.isnull().sum()
```

Out[6]:

```
Port Number                      0
Received Packets                 0
Received Bytes                   0
Sent Bytes                       0
Sent Packets                     0
Port alive Duration (S)          0
Packets Rx Dropped               0
Packets Tx Dropped               0
Packets Rx Errors                0
Packets Tx Errors                0
Delta Received Packets           0
Delta Received Bytes             0
Delta Sent Bytes                 0
Delta Sent Packets               0
Delta Port alive Duration (S)    0
Delta Packets Rx Dropped         0
Delta Packets Tx Dropped         0
Delta Packets Rx Errors          0
Delta Packets Tx Errors          0
Connection Point                 0
Total Load/Rate                  0
Total Load/Latest                0
Unknown Load/Rate                0
Unknown Load/Latest              0
Latest bytes counter             0
is_valid                         0
Table ID                         0
Active Flow Entries              0
Packets Looked Up                0
Packets Matched                  0
Max Size                         0
Label                            0
dtype: int64
```

Detailed description of the dataset

In [7]:

```python
data_frame.describe()
```

Out[7]:

	Port Number	Received Packets	Received Bytes	Sent Bytes	Sent Packets	D
count	4927.000000	4927.000000	4.927000e+03	4.927000e+03	4927.000000	49
mean	2.237061	85133.248427	4.776574e+07	4.798439e+07	150534.093363	13
std	1.063085	122860.550433	4.963905e+07	4.906864e+07	149729.243633	9
min	1.000000	10.000000	8.560000e+02	5.775000e+03	42.000000	
25%	1.000000	875.000000	1.170596e+07	1.029537e+07	1106.500000	2
50%	2.000000	3721.000000	2.674802e+07	3.109155e+07	151603.000000	14
75%	3.000000	179378.000000	7.574614e+07	7.956961e+07	288375.500000	22
max	4.000000	352772.000000	2.715916e+08	2.392430e+08	421598.000000	33

8 rows × 32 columns

Target label and its count

In [8]:

```
data_frame["Label"].value_counts()
```

Out[8]:

```
Label
0    2641
3     656
2     646
1     589
4     395
Name: count, dtype: int64
```

Since, label 0 alone contains more sample, perform under sampling

In [9]:

```
label_0_indices = data_frame[data_frame["Label"] == 0].index
```

Under Sampling

In [10]:

```
indices_to_remove = np.random.choice(
    label_0_indices,
    size=2000,
    replace=False,
)
data_frame = data_frame.drop(indices_to_remove)
```

Label count after Under Sampling

In [11]:

```
data_frame["Label"].value_counts()
```

Out[11]:

```
Label
3    656
2    646
0    641
1    589
4    395
Name: count, dtype: int64
```

Find the same valued columns

In [12]:

```
for column in data_frame.columns:
    column_max_value = max(data_frame[column])
    column_min_value = min(data_frame[column])

    # If max and min are same for current column
    if column_max_value == column_min_value:
        print(column)  # Print dropped columns
        data_frame.drop(
            column,
            axis=1,
            inplace=True,
        )  # Drop the current column

print("Same valued columns had been dropped from the data frame.")
```

```
Packets Rx Dropped
Packets Tx Dropped
Packets Rx Errors
Packets Tx Errors
Delta Packets Rx Dropped
Delta Packets Tx Dropped
Delta Packets Rx Errors
Delta Packets Tx Errors
is_valid
Table ID
Max Size
Same valued columns had been dropped from the data frame.
```

In [13]:

```
data_frame.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 2927 entries, 1 to 4926
Data columns (total 21 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   Port Number                     2927 non-null   int64
 1   Received Packets                2927 non-null   int64
 2   Received Bytes                  2927 non-null   int64
 3   Sent Bytes                      2927 non-null   int64
 4   Sent Packets                    2927 non-null   int64
 5   Port alive Duration (S)         2927 non-null   int64
 6   Delta Received Packets          2927 non-null   int64
 7   Delta Received Bytes            2927 non-null   int64
 8   Delta Sent Bytes                2927 non-null   int64
 9   Delta Sent Packets              2927 non-null   int64
 10  Delta Port alive Duration (S)   2927 non-null   int64
 11  Connection Point                2927 non-null   int64
 12  Total Load/Rate                 2927 non-null   int64
 13  Total Load/Latest               2927 non-null   int64
 14  Unknown Load/Rate               2927 non-null   int64
 15  Unknown Load/Latest             2927 non-null   int64
 16  Latest bytes counter            2927 non-null   int64
 17  Active Flow Entries             2927 non-null   int64
 18  Packets Looked Up               2927 non-null   int64
 19  Packets Matched                 2927 non-null   int64
 20  Label                           2927 non-null   int64
dtypes: int64(21)
memory usage: 503.1 KB
```

Find the correlation between each column

In [14]:

```
columns_to_correlate = data_frame.iloc[:, :-1]
columns_to_correlate.corr()
```

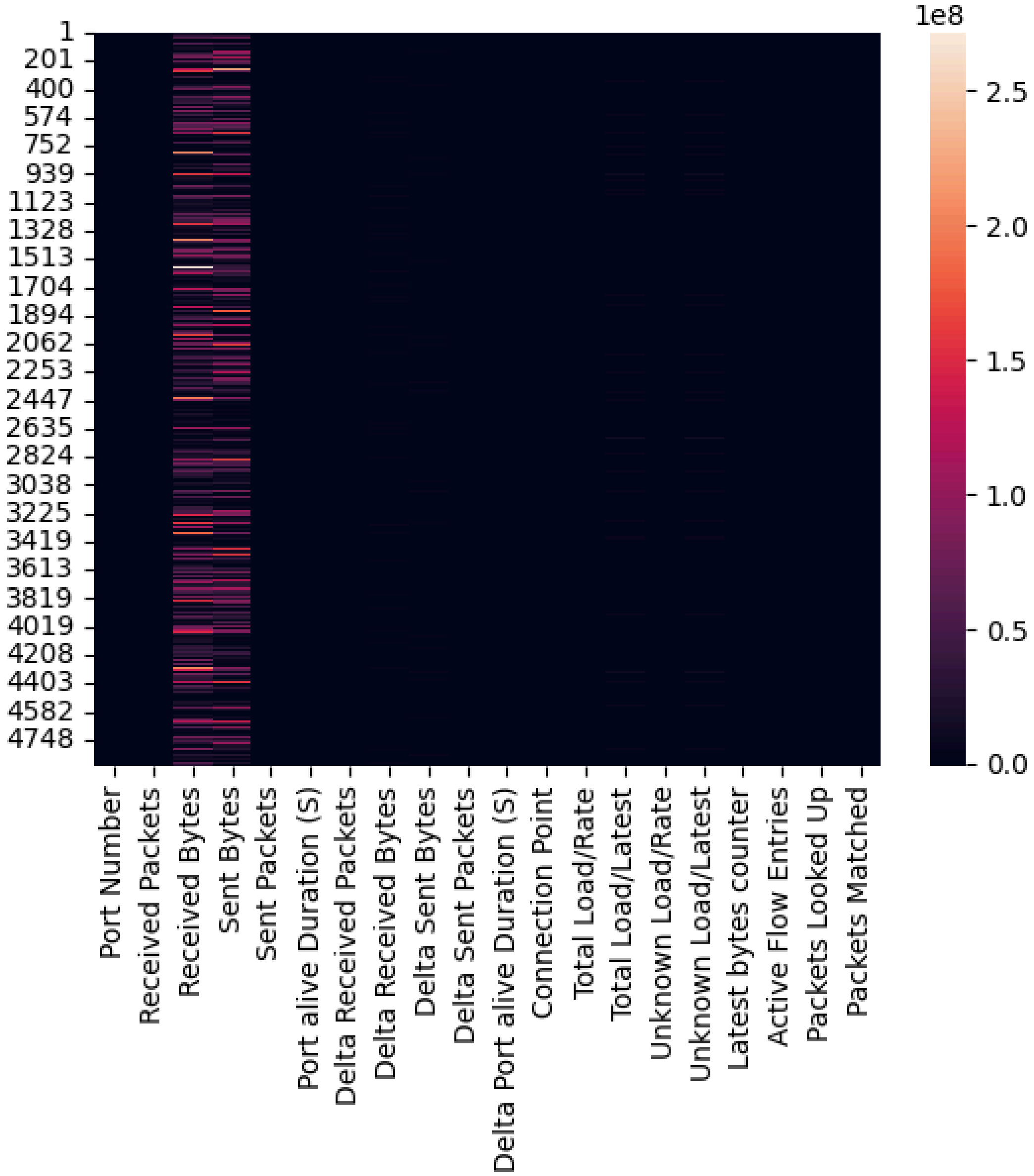
Out[14]:

	Port Number	Received Packets	Received Bytes	Sent Bytes	Sent Packets	Port alive Duration (S)	Rec Pa
Port Number	1.000000	0.252370	-0.034394	0.025225	-0.041110	0.017693	0.00
Received Packets	0.252370	1.000000	0.235834	0.308183	0.534474	0.279687	0.1
Received Bytes	-0.034394	0.235834	1.000000	0.664012	0.449018	0.748623	-0.04
Sent Bytes	0.025225	0.308183	0.664012	1.000000	0.568606	0.743222	-0.07
Sent Packets	-0.041110	0.534474	0.449018	0.568606	1.000000	0.387870	-0.00
Port alive Duration (S)	0.017693	0.279687	0.748623	0.743222	0.387870	1.000000	-0.12
Delta Received Packets	0.003846	0.101214	-0.049997	-0.074865	-0.006359	-0.127957	1.00
Delta Received Bytes	0.020309	-0.011130	0.119992	0.008524	0.028972	0.024346	0.0
Delta Sent Bytes	-0.046947	-0.014515	0.031163	0.098878	0.021135	0.036452	0.06
Delta Sent Packets	-0.026259	0.039652	-0.066122	-0.045771	0.043975	-0.117409	0.66
Delta Port alive Duration (S)	0.011344	-0.083028	-0.048477	-0.087523	-0.137544	-0.043869	0.02
Connection Point	0.908395	0.228202	0.061944	0.010040	-0.066948	0.131502	-0.01
Total Load/Rate	0.030156	0.039386	0.077851	0.059491	0.043196	0.032990	0.0
Total Load/Latest	0.055210	0.071181	0.049543	0.054790	0.038828	0.004888	-0.00
Unknown Load/Rate	0.030156	0.039386	0.077851	0.059491	0.043196	0.032990	0.0
Unknown Load/Latest	0.055210	0.071181	0.049543	0.054790	0.038828	0.004888	-0.00
Latest bytes counter	0.030156	0.039386	0.077851	0.059491	0.043196	0.032990	0.0

	Port Number	Received Packets	Received Bytes	Sent Bytes	Sent Packets	Port alive Duration (S)	Rec Pa
Active Flow Entries	0.008266	-0.076747	0.022085	0.015508	-0.098111	0.129453	-0.02
Packets Looked Up	0.037100	0.718313	0.436735	0.531854	0.937647	0.396722	-0.0
Packets Matched	0.037090	0.718313	0.436679	0.531810	0.937646	0.396645	-0.0

Plot the heatmap for better visualization

In [15]:
```
sbn.heatmap(columns_to_correlate)
mpl.show()
```



The above heatmap is incorrect. Since the range of the column values differs.

The Min-Max Normalization is to be applied to get the correct correlation

In [16]:

```
# Scaling
scaler = MinMaxScaler()
df_scaled = scaler.fit_transform(columns_to_correlate.to_numpy())

new_data_frame = pd.DataFrame(
    df_scaled,
    columns=columns_to_correlate.columns,
)
```

Find the scaled Min-Max values for future transformation

In [17]:

```
scaled_min = scaler.data_min_
print(scaled_min)
scaled_max = scaler.data_max_
print(scaled_max)
```

```
[ 1.00000e+00  1.00000e+01  8.56000e+02  5.77500e+03  4.20000e+01
  2.60000e+01  0.00000e+00  0.00000e+00  2.78000e+02  2.00000e+00
  4.00000e+00  1.00000e+00 -6.30355e+05  0.00000e+00 -6.30355e+05
  0.00000e+00 -6.30355e+05  4.00000e+00  1.05000e+02  5.00000e+01]
[4.00000000e+00 3.52772000e+05 2.71591638e+08 2.39233335e+08
 4.21315000e+05 3.31700000e+03 1.56590000e+04 6.30270800e+06
 6.30270800e+06 1.55920000e+04 5.00000000e+00 5.00000000e+00
 1.74674900e+06 1.74674920e+07 1.74674900e+06 1.74674920e+07
 1.74674900e+06 6.08000000e+02 1.01156300e+06 1.01142800e+06]
```

Correlation after scaling

In [18]:

```
new_data_frame.corr()
```

Out[18]:

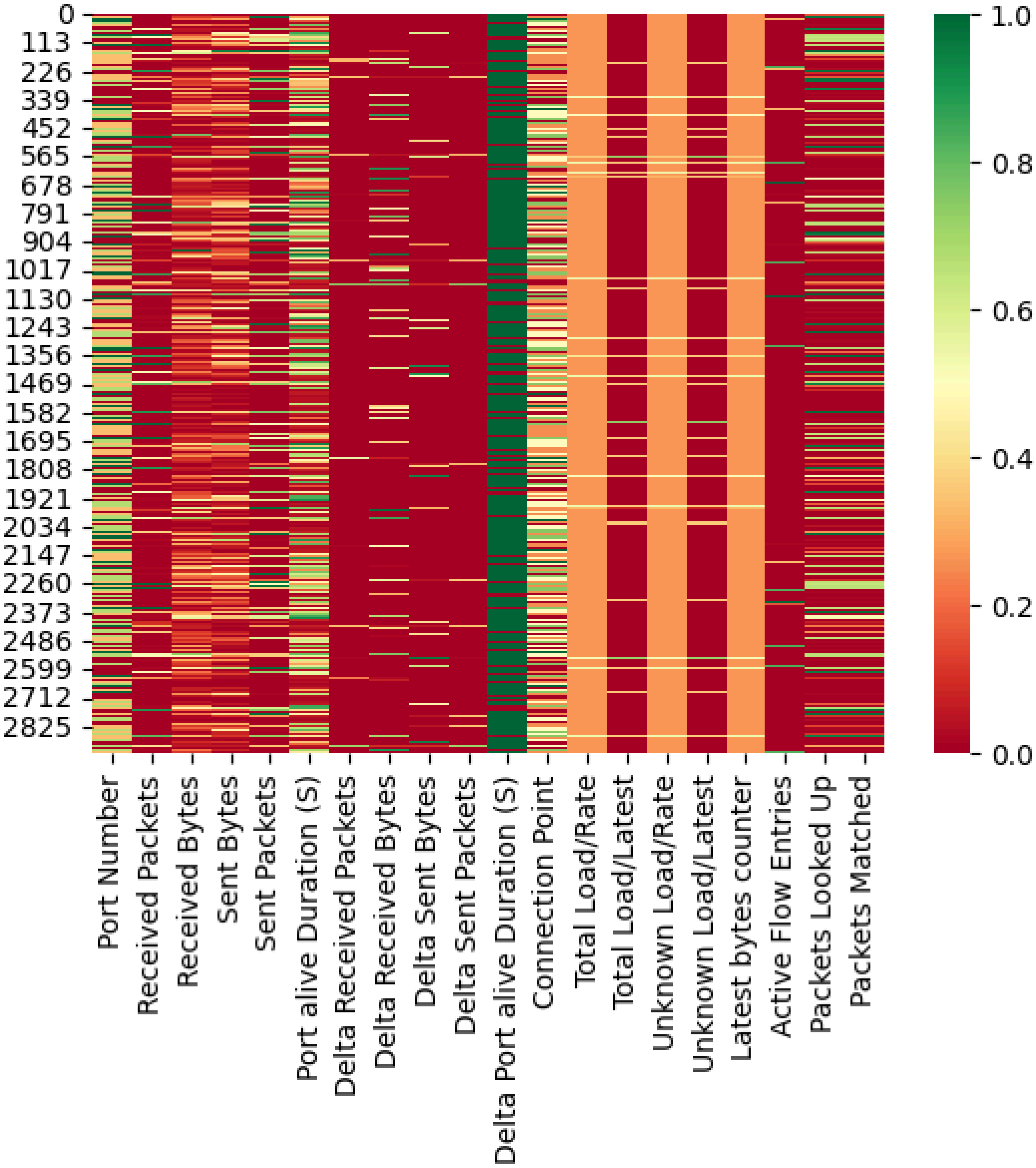
	Port Number	Received Packets	Received Bytes	Sent Bytes	Sent Packets	Port alive Duration (S)	Rec Pa
Port Number	1.000000	0.252370	-0.034394	0.025225	-0.041110	0.017693	0.00
Received Packets	0.252370	1.000000	0.235834	0.308183	0.534474	0.279687	0.1
Received Bytes	-0.034394	0.235834	1.000000	0.664012	0.449018	0.748623	-0.04
Sent Bytes	0.025225	0.308183	0.664012	1.000000	0.568606	0.743222	-0.07
Sent Packets	-0.041110	0.534474	0.449018	0.568606	1.000000	0.387870	-0.00
Port alive Duration (S)	0.017693	0.279687	0.748623	0.743222	0.387870	1.000000	-0.12
Delta Received Packets	0.003846	0.101214	-0.049997	-0.074865	-0.006359	-0.127957	1.00
Delta Received Bytes	0.020309	-0.011130	0.119992	0.008524	0.028972	0.024346	0.07
Delta Sent Bytes	-0.046947	-0.014515	0.031163	0.098878	0.021135	0.036452	0.06
Delta Sent Packets	-0.026259	0.039652	-0.066122	-0.045771	0.043975	-0.117409	0.66
Delta Port alive Duration (S)	0.011344	-0.083028	-0.048477	-0.087523	-0.137544	-0.043869	0.02
Connection Point	0.908395	0.228202	0.061944	0.010040	-0.066948	0.131502	-0.01
Total Load/Rate	0.030156	0.039386	0.077851	0.059491	0.043196	0.032990	0.01
Total Load/Latest	0.055210	0.071181	0.049543	0.054790	0.038828	0.004888	-0.00
Unknown Load/Rate	0.030156	0.039386	0.077851	0.059491	0.043196	0.032990	0.01
Unknown Load/Latest	0.055210	0.071181	0.049543	0.054790	0.038828	0.004888	-0.00
Latest bytes counter	0.030156	0.039386	0.077851	0.059491	0.043196	0.032990	0.01

	Port Number	Received Packets	Received Bytes	Sent Bytes	Sent Packets	Port alive Duration (S)	Rec Pa
Active Flow Entries	0.008266	-0.076747	0.022085	0.015508	-0.098111	0.129453	-0.02
Packets Looked Up	0.037100	0.718313	0.436735	0.531854	0.937647	0.396722	-0.0
Packets Matched	0.037090	0.718313	0.436679	0.531810	0.937646	0.396645	-0.0

Heatmap after scaling

In [19]:

```
sbn.heatmap(
    new_data_frame,
    cmap="RdYlGn",
)
mpl.show()
```



Now, we can clearly see that before scaling the heatmap base was le8.
After scaling the base for heatmap had been changed to 0 - 1

Drop the least correlated columns

```
In [20]: updated_data_frame = new_data_frame.drop(
    [
        "Delta Received Packets",
        "Delta Sent Packets",
        "Total Load/Latest",
        "Unknown Load/Rate",
        "Unknown Load/Latest",
        "Latest bytes counter",
        "Packets Looked Up",
    ],
    axis=1,
)
# [6, 9, 13, 14, 15, 16, 18] indices with respect to updated dataframe
```

Drop the least correlated columns in scaled min, max values

In [21]:

```
index_locations_to_remove = [6, 9, 13, 14, 15, 16, 18]

# Remove items at the specified index locations
scaled_min_filtered = [
    scaled_min[i] for i in range(len(scaled_min)) if i not in index_locatior
]

scaled_max_filtered = [
    scaled_max[i] for i in range(len(scaled_max)) if i not in index_locatior
]

print("Filtered scaled_min list:")
print(scaled_min_filtered)

print("\nFiltered scaled_max list:")
print(scaled_max_filtered)
```

```
Filtered scaled_min list:
[1.0, 10.0, 856.0, 5775.0, 42.0, 26.0, 0.0, 278.0, 4.0, 1.0, -630355.0, 4.0,
50.0]

Filtered scaled_max list:
[4.0, 352772.0, 271591638.0, 239233335.0, 421315.0, 3317.0, 6302708.0, 63027
08.0, 5.0, 5.0, 1746749.0, 608.0, 1011428.0]
```

Updated dataframe

In [22]:

```
updated_data_frame.head()
```

Out[22]:

	Port Number	Received Packets	Received Bytes	Sent Bytes	Sent Packets	Port alive Duration (S)	Delta Received Bytes	Delta Sent Bytes
0	0.333333	0.000564	0.000073	0.026380	0.000551	0.021270	0.000560	0.437557
1	1.000000	0.000397	0.000070	0.027044	0.007150	0.042540	0.000088	0.000919
2	0.666667	0.002778	0.000380	0.000347	0.001747	0.232756	0.000088	0.000044
3	0.000000	0.000995	0.046419	0.000102	0.000686	0.028867	0.000000	0.000055
4	0.333333	0.004884	0.139417	0.159087	0.007465	0.646004	0.000000	0.000044

In [23]:

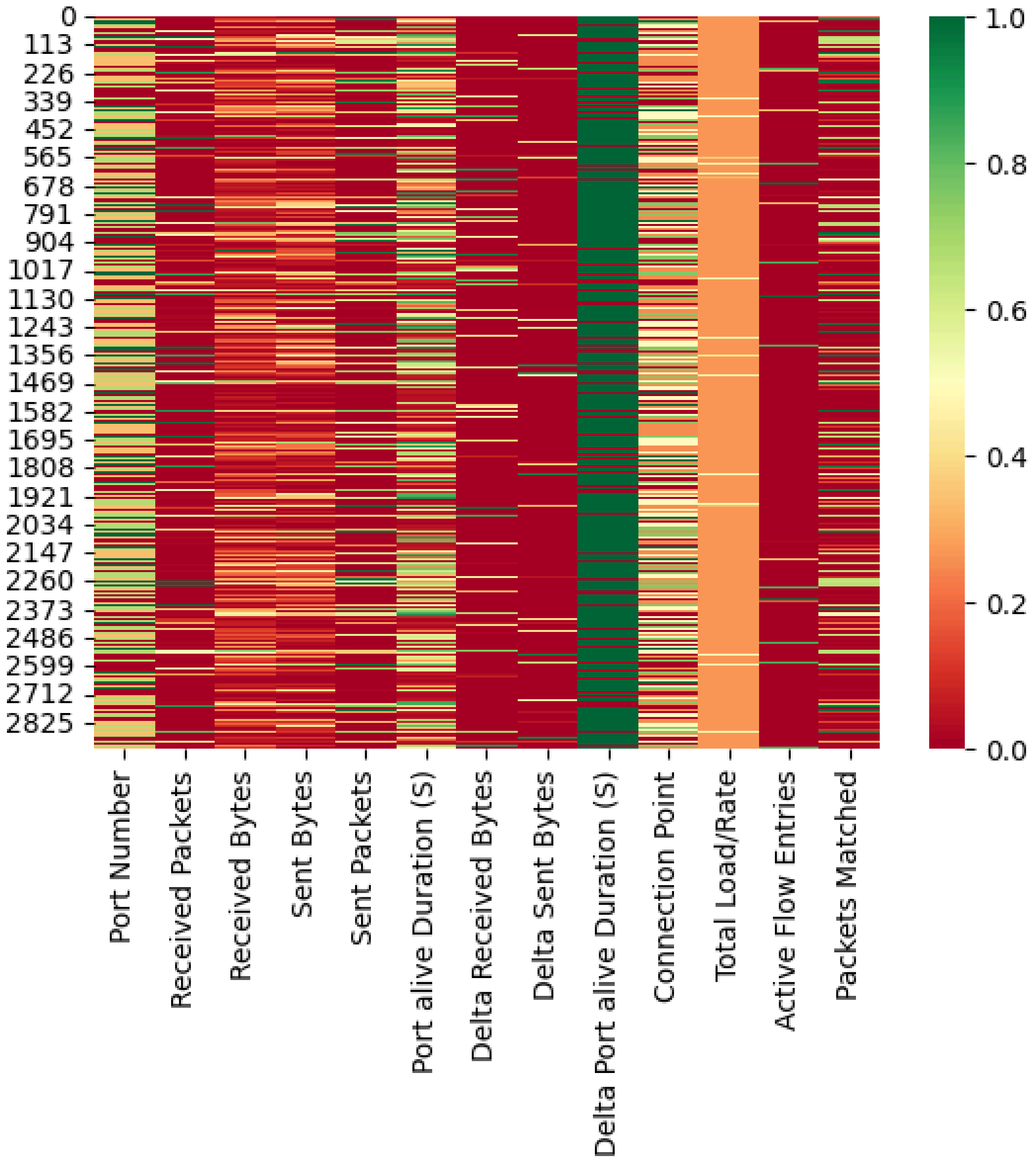
```
updated_data_frame.describe()
```

Out[23]:

	Port Number	Received Packets	Received Bytes	Sent Bytes	Sent Packets	Port ali Duration (
count	2927.000000	2927.000000	2927.000000	2927.000000	2927.000000	2927.00000
mean	0.405876	0.109599	0.121487	0.128869	0.151822	0.3137
std	0.347302	0.253964	0.158419	0.166508	0.282056	0.3080
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.0000
25%	0.000000	0.001140	0.004890	0.000597	0.000968	0.0379
50%	0.333333	0.004819	0.054604	0.053287	0.004494	0.2123
75%	0.666667	0.013423	0.168076	0.185447	0.100415	0.5828
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.0000

In [24]:

```
sbn.heatmap(
    updated_data_frame,
    cmap="RdYlGn",
)
mpl.show()
```



Append the label column to the updated data frame

In [25]:

```
updated_data_frame["Label"] = data_frame["Label"].values
```

In [26]:

```
updated_data_frame.head()
```

Out[26]:

	Port Number	Received Packets	Received Bytes	Sent Bytes	Sent Packets	Port alive Duration (S)	Delta Received Bytes	Delta Sent Bytes
0	0.333333	0.000564	0.000073	0.026380	0.000551	0.021270	0.000560	0.437557
1	1.000000	0.000397	0.000070	0.027044	0.007150	0.042540	0.000088	0.000919
2	0.666667	0.002778	0.000380	0.000347	0.001747	0.232756	0.000088	0.000044
3	0.000000	0.000995	0.046419	0.000102	0.000686	0.028867	0.000000	0.000055
4	0.333333	0.004884	0.139417	0.159087	0.007465	0.646004	0.000000	0.000044

Save the new dataframe to csv for modelling

In [27]:

```
updated_data_frame.to_csv(
    "../datasets/cleaned_dataset.csv",
    index=False,
    header=True,
)
print("Preprocessed dataset saved successfully.")
```

```
Preprocessed dataset saved successfully.
```