## Libraries and paths

In [11]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

models = ['models_2019_06_01_21_21/',
          'models_2019_06_01_21_44/',
          'models_2019_06_01_22_11/']
```

## Load data

In [12]:
```
data_30T = pd.read_csv(models[0] + 'detected_anomalies.csv',
                       sep='\t',
                       index_col=[0],)
data_10T = pd.read_csv(models[1] + 'detected_anomalies.csv',
                       sep='\t',
                       index_col=[0],)
data_5T = pd.read_csv(models[2] + 'detected_anomalies.csv',
                      sep='\t',
                      index_col=[0],)
print('Anomalies loaded!')
```

```
Anomalies loaded!
```

In [13]:
```
data_30T.head()
```

Out[13]:

| | ('avg_Length', 'avg_2019-02-01 09:30:00') | ('avg_Length', 'avg_2019-02-01 10:00:00') | ('avg_Length', 'avg_2019-02-01 10:30:00') | ('avg_Length', 'avg_2019-02-01 11:00:00') | ('avg_Length', 'avg_2019-02-01 11:30:00') | ('avg_Length', 'avg_2019-02-01 12:00:00') | ('avg_Le 'avg_20 12:30:0 |
|---|---|---|---|---|---|---|---|
| **104.192.1.15source** | 60.000000 | 59.978492 | 59.802326 | 60.000000 | 60.854239 | 60.613653 | 60 |
| **104.248.135.165source** | 61.504604 | 61.425604 | 61.297026 | 61.551904 | 61.367338 | 61.300701 | 61 |
| **104.248.37.237source** | 61.878968 | 61.898538 | 61.447473 | 59.957901 | 60.179832 | 59.987654 | 60 |
| **122.228.19.80source** | 44.000000 | 44.000000 | 44.000000 | 44.000000 | 44.000000 | 44.000000 | 44 |
| **133.34.156.192source** | 40.000000 | 40.031746 | 40.000000 | 108.638418 | 105.238342 | 40.000000 | 157 |

5 rows × 1102 columns

In [14]:
```
data_10T.head()
```

Out[14]:

| | ('avg_Length', 'avg_2019-02-01 09:30:00') | ('avg_Length', 'avg_2019-02-01 09:40:00') | ('avg_Length', 'avg_2019-02-01 09:50:00') | ('avg_Length', 'avg_2019-02-01 10:00:00') | ('avg_Length', 'avg_2019-02-01 10:10:00') | ('avg_Length', 'avg_2019-02-01 10:20:00') | ('avg_Le 'avg_20 10:30:0 |
|---|---|---|---|---|---|---|---|
| **104.192.1.15source** | 60.000000 | 60.000000 | 60.000000 | 60.181121 | 60.134423 | 59.200262 | 59 |
| **104.248.135.165source** | 60.000000 | 60.000000 | 61.549547 | 61.269025 | 61.802870 | 61.192314 | 61 |
| **104.248.37.237source** | 63.946527 | 61.741286 | 61.572278 | 61.074691 | 62.175765 | 62.102029 | 61 |
| **122.228.19.80source** | 44.000000 | 44.000000 | 44.000000 | 44.000000 | 44.000000 | 44.000000 | ( |
| **133.34.156.192source** | 40.000000 | 40.000000 | 40.000000 | 40.090909 | 40.000000 | 40.000000 | 40 |

5 rows × 3202 columns

In [19]:
```
data_5T.head()
```

Out[19]:

| | ('avg_Length', 'avg_2019-02-01 09:35:00') | ('avg_Length', 'avg_2019-02-01 09:40:00') | ('avg_Length', 'avg_2019-02-01 09:45:00') | ('avg_Length', 'avg_2019-02-01 09:50:00') | ('avg_Length', 'avg_2019-02-01 09:55:00') | ('avg_Length', 'avg_2019-02-01 10:00:00') | ('avg_Le 'avg_20 10:05:0 |
|---|---|---|---|---|---|---|---|
| **103.9.179.179source** | 0.000000 | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0 |
| **104.192.1.15source** | 60.000000 | 60.000000 | 60.00000 | 60.000000 | 60.000000 | 60.000000 | 60 |
| **104.248.135.165source** | 60.000000 | 60.000000 | 60.00000 | 61.557567 | 61.542310 | 61.398975 | 61 |
| **104.248.228.27source** | 40.000000 | 40.000000 | 0.00000 | 50.500000 | 53.000000 | 44.000000 | 40 |
| **104.248.37.237source** | 63.946527 | 61.981725 | 61.43956 | 61.983410 | 61.127975 | 61.215415 | 60 |

5 rows × 6310 columns

In [20]:
```
print(f'Shape of data_30T is {data_30T.shape}')
print(f'Shape of data_10T is {data_10T.shape}')
print(f'Shape of data_5T is {data_5T.shape}')
```

```
Shape of data_30T is (368, 1102)
Shape of data_10T is (387, 3202)
Shape of data_5T is (396, 6310)
```

In [21]:
```
common_30T_5T = data_5T.index.isin(data_30T.index)
common_10T_5T = data_5T.index.isin(data_10T.index)
common_three = common_30T_5T & common_10T_5T
print(f'Number of common anomalies between 30T and 5T is {common_30T_5T.sum()}')
print(f'Number of common anomalies between 10T and 5T is {common_10T_5T.sum()}')
print(f'Number of common anomalies between all three is {common_three.sum()}')
```

```
Number of common anomalies between 30T and 5T is 362
Number of common anomalies between 10T and 5T is 383
Number of common anomalies between all three is 360
```

- Most anomalies are common between all three
- A hard look on 5T would give most of the insight for all three
- special cases might be found due to the window period
- Special cases must be looked at

## Main evaluation will be performed on 5T dataset

In [22]:
```
data_5T_full = pd.read_csv('../data/final/final_dataset_5T.csv',
                           sep='\t',
                           index_col=[0])
data_5T = data_5T.drop(['host0','host1', 'host2',
                        'host3', 'host4', 'host5',
                        'host6', 'host7', 'is_src',
                        'is_anomaly'],
                       axis=1)
data_5T_full = data_5T_full.drop(['host0','host1', 'host2',
                        'host3', 'host4', 'host5',
                        'host6', 'host7', 'is_src'],
                       axis=1)
```

In [46]:
```
data_5T.columns = data_5T.columns.map(eval)
data_5T_full.columns = data_5T.columns.map(eval)
```

In [47]:
```
data_5T.head()
```

Out[47]:

| | avg_Length | | | | | | |
|---|---|---|---|---|---|---|---|
| | avg_2019-02-01 09:35:00 | avg_2019-02-01 09:40:00 | avg_2019-02-01 09:45:00 | avg_2019-02-01 09:50:00 | avg_2019-02-01 09:55:00 | avg_2019-02-01 10:00:00 | avg_2019-0 10:05:00 |
| **103.9.179.179source** | 0.000000 | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.00 |
| **104.192.1.15source** | 60.000000 | 60.000000 | 60.00000 | 60.000000 | 60.000000 | 60.000000 | 60.38 |
| **104.248.135.165source** | 60.000000 | 60.000000 | 60.00000 | 61.557567 | 61.542310 | 61.398975 | 61.07 |
| **104.248.228.27source** | 40.000000 | 40.000000 | 0.00000 | 50.500000 | 53.000000 | 44.000000 | 40.00 |
| **104.248.37.237source** | 63.946527 | 61.981725 | 61.43956 | 61.983410 | 61.127975 | 61.215415 | 60.94 |

5 rows × 6300 columns

In [138]:
```
data_5T_full.head()
```

Out[138]:

| | avg_Length | | | | | | |
|---|---|---|---|---|---|---|---|
| | avg_2019-02-01 09:35:00 | avg_2019-02-01 09:40:00 | avg_2019-02-01 09:45:00 | avg_2019-02-01 09:50:00 | avg_2019-02-01 09:55:00 | avg_2019-02-01 10:00:00 | avg_2019-02-01 10:05:00 |
| **1.0.171.245source** | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| **1.0.255.54source** | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| **1.1.158.248source** | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| **1.1.217.60source** | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| **1.1.230.15source** | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |

5 rows × 6300 columns

Get rid of the anomalies and generate a sample of 4000 (around 10%)

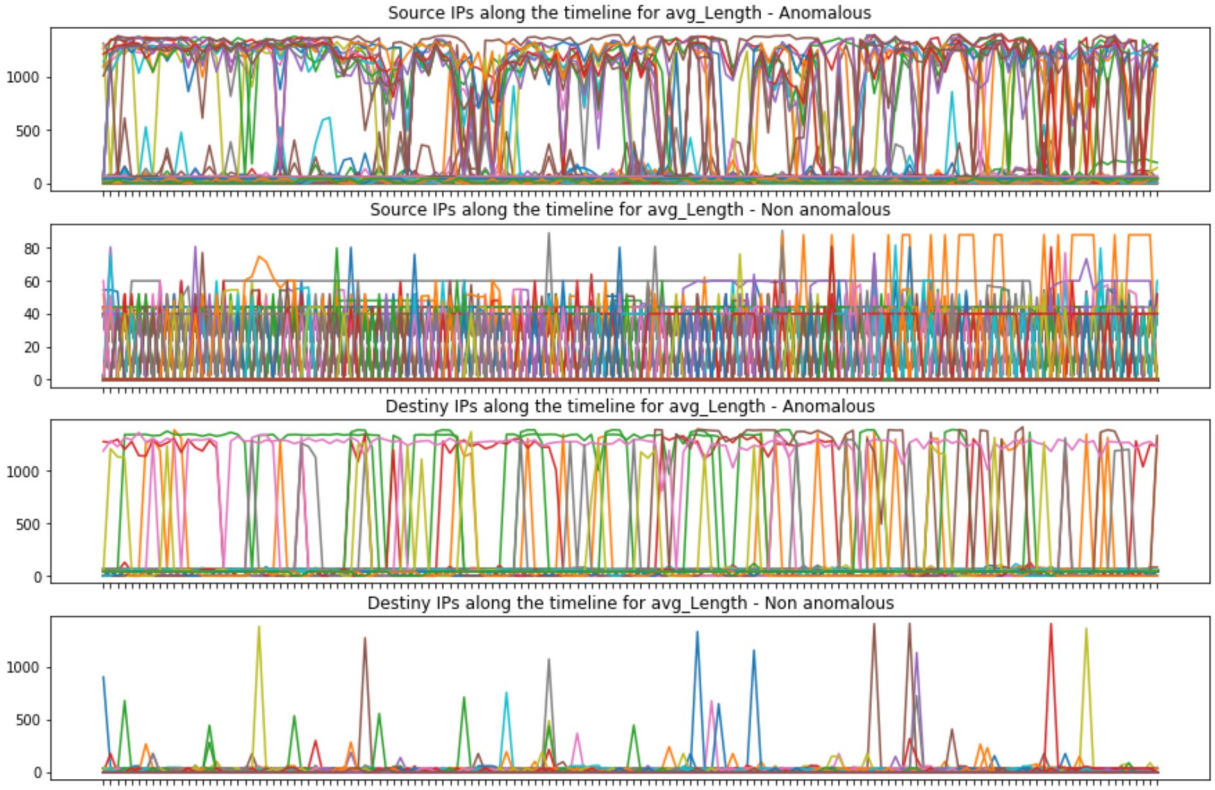
In [149]:
```
idx = ~data_5T_full.index.isin(data_5T.index)
data_5T_full = data_5T_full[idx]
data_5T_sample = data_5T_full.sample(4000)
```

In [150]:
```
def plot_all_ips(data, sample, var):
    fig, ax = plt.subplots(nrows=4, figsize=(15,10))
    source_ips = [c for c in data.index if 'source' in c]
    sample_source = [c for c in sample.index if 'source' in c]
    dst_ips = [c for c in data.index if 'dst' in c]
    sample_dst = [c for c in sample.index if 'dst' in c]
    for i in source_ips:
        ax[0].plot(data.loc[i, var])
    for i in sample_source:
        ax[1].plot(sample.loc[i, var])
    for j in dst_ips:
        ax[2].plot(data.loc[j, var])
    for j in sample_dst:
        ax[3].plot(sample.loc[j, var])

    ax[0].set_title(f'Source IPs along the timeline for {var} - Anomalous')
    ax[0].set_xticklabels([])
    ax[1].set_title(f'Source IPs along the timeline for {var} - Non anomalous')
    ax[1].set_xticklabels([])
    ax[2].set_title(f'Destiny IPs along the timeline for {var} - Anomalous')
    ax[2].set_xticklabels([])
    ax[3].set_title(f'Destiny IPs along the timeline for {var} - Non anomalous')
    ax[3].set_xticklabels([])
```

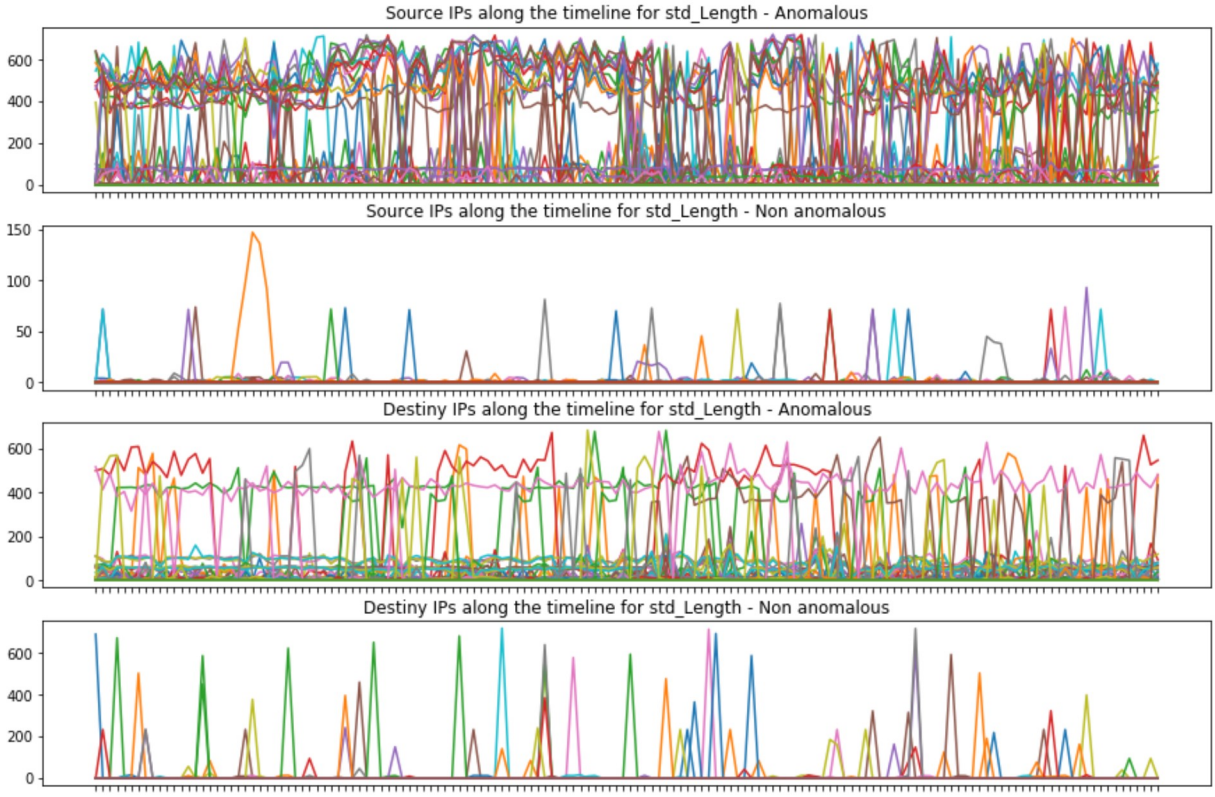
# Length

In [152]:
```
plot_all_ips(data_5T, data_5T_sample, 'avg_Length')
```



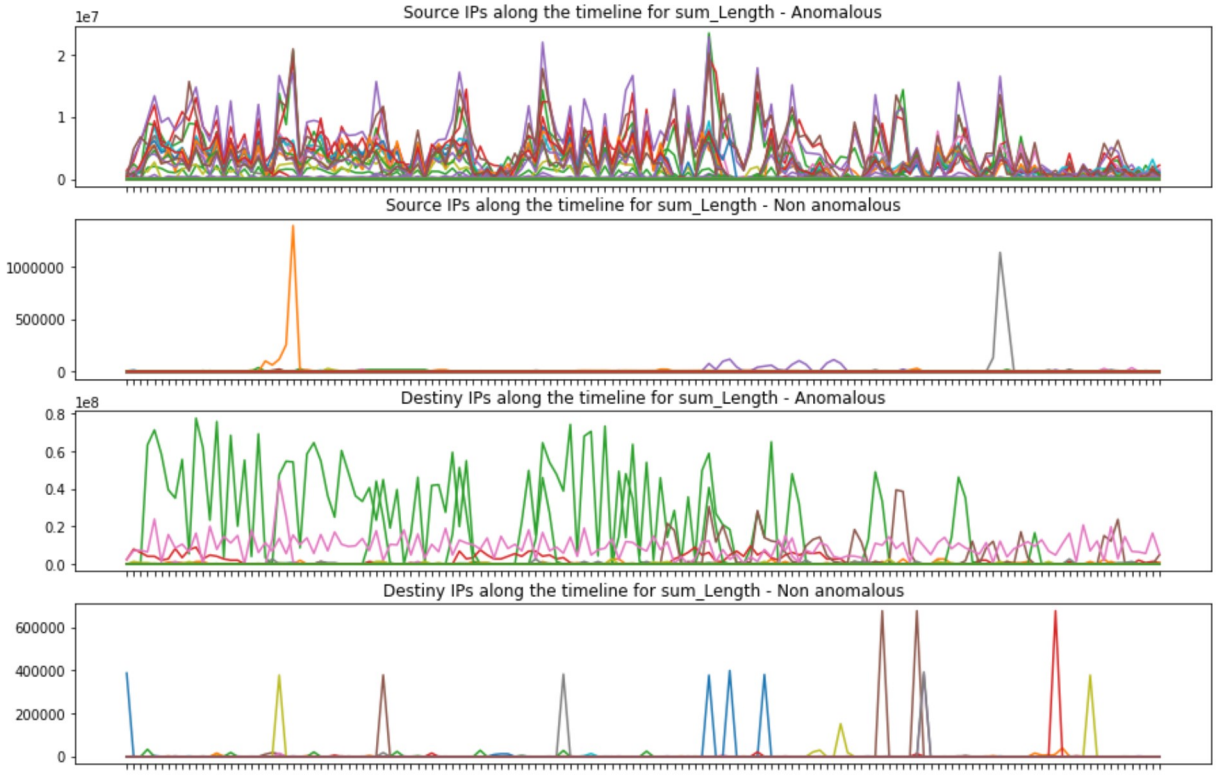

- Most anomalous source ips have a high variation of packet avg_length
- Destiny ips avg_length is lower than sources ips
- Average length is much higher for anomalous ips

In [153]:
```
plot_all_ips(data_5T, data_5T_sample, 'std_Length')
```



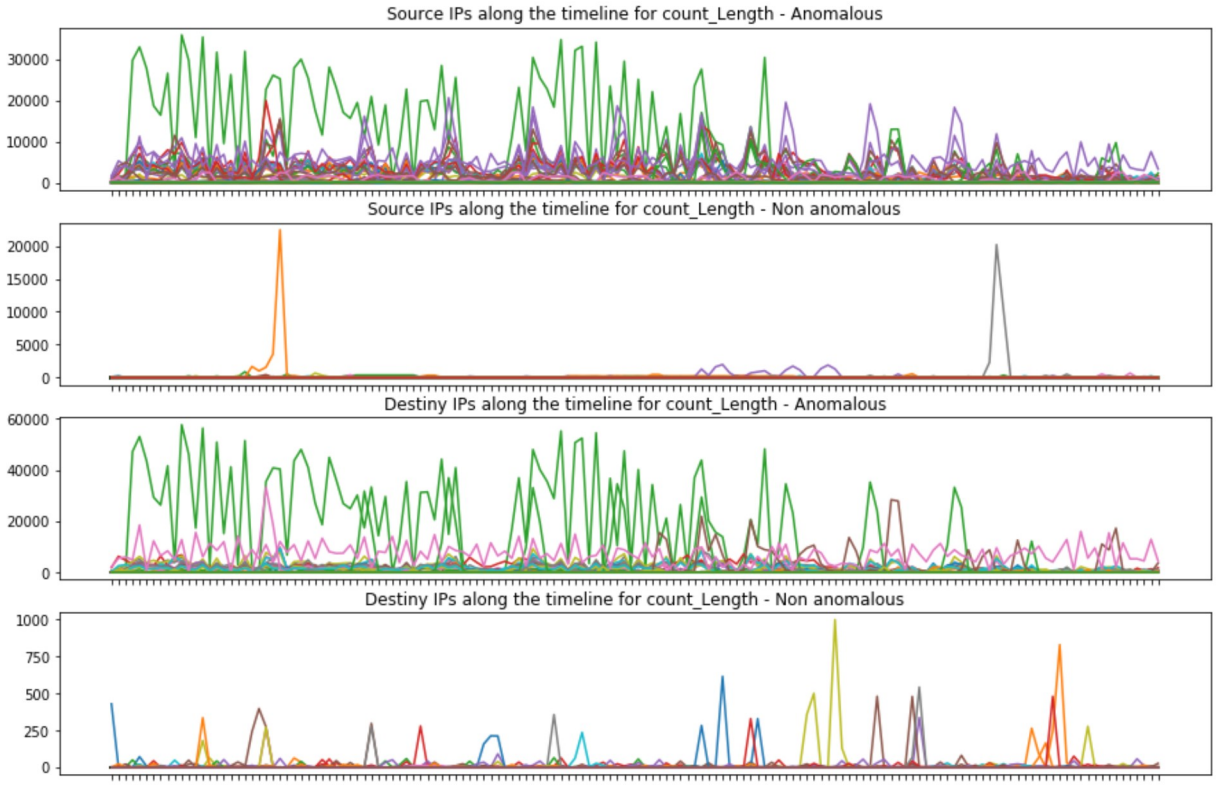

- Most source ips have either a great std or really low
- For destiny ips most ips tends towards a lower std, meaning that most destiny ips have lower packages lengths overall
- Even though most non anomalous ips have a really low average, there is still some std for destiny non anomalous ips

In [154]:
```
plot_all_ips(data_5T, data_5T_sample, 'sum_Length')
```



- Similar trend for mos source ips
- In destiny ips there is three above the rest, for the most part the sum is pretty low
- Sum of the packet lengths is much higher and variate in anomalous than non anomalous

In [155]:
```
plot_all_ips(data_5T, data_5T_sample, 'count_Length')
```




- Both for source and destiny there are a couple of ips that stand out
- Those ips have the same form, so there is a connection between two ips (source and dst) that really stand out from both ends

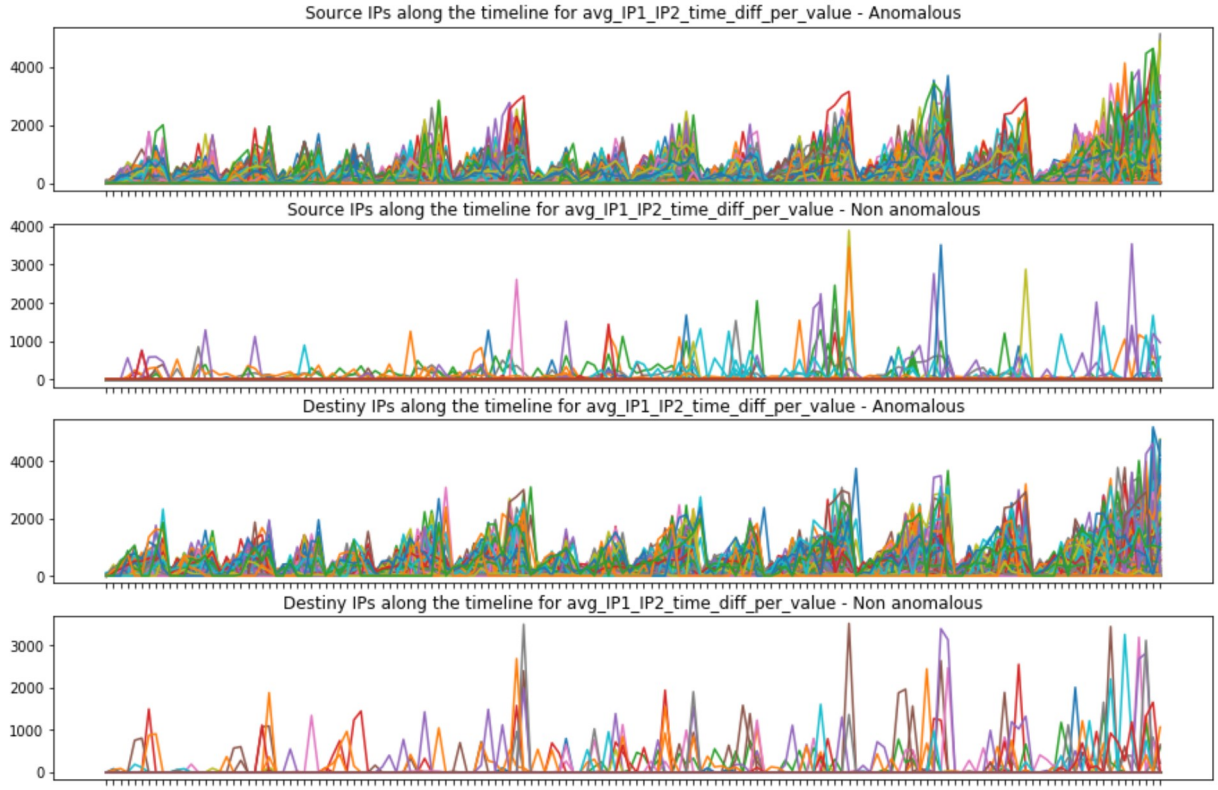
*TODO: get those singular ips*

# Length summary

- Higher package length in anomalous
- Anomalous and non anomalous have similar std, whereas for source ips the std in non anomalous is pretty low
- Amount of information transmited (sum) is much higher in anomalous IPs
- Anomalies have higher counts, which means higher connections during the timeline
- The difference in package length could point to malware inyection

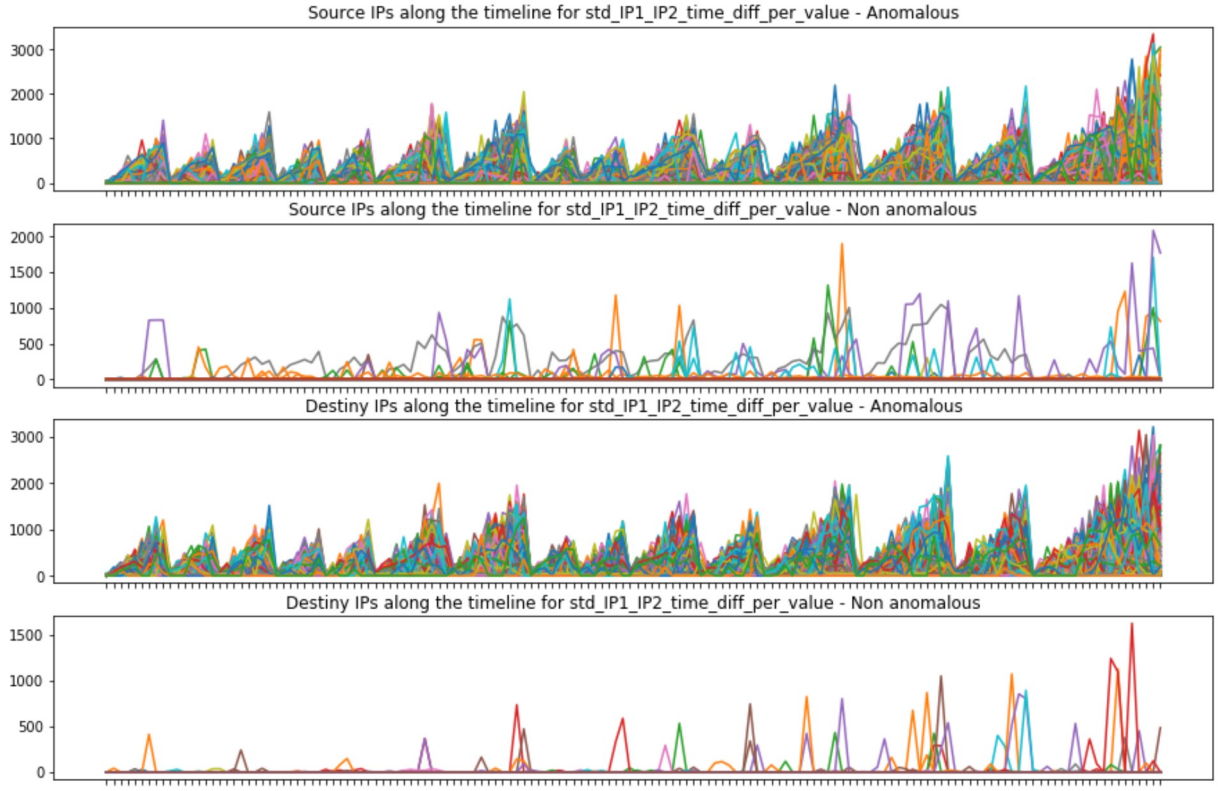
# IP1_IP2_time_diff_per_value

In [156]:
```
plot_all_ips(data_5T, data_5T_sample, 'avg_IP1_IP2_time_diff_per_value')
```



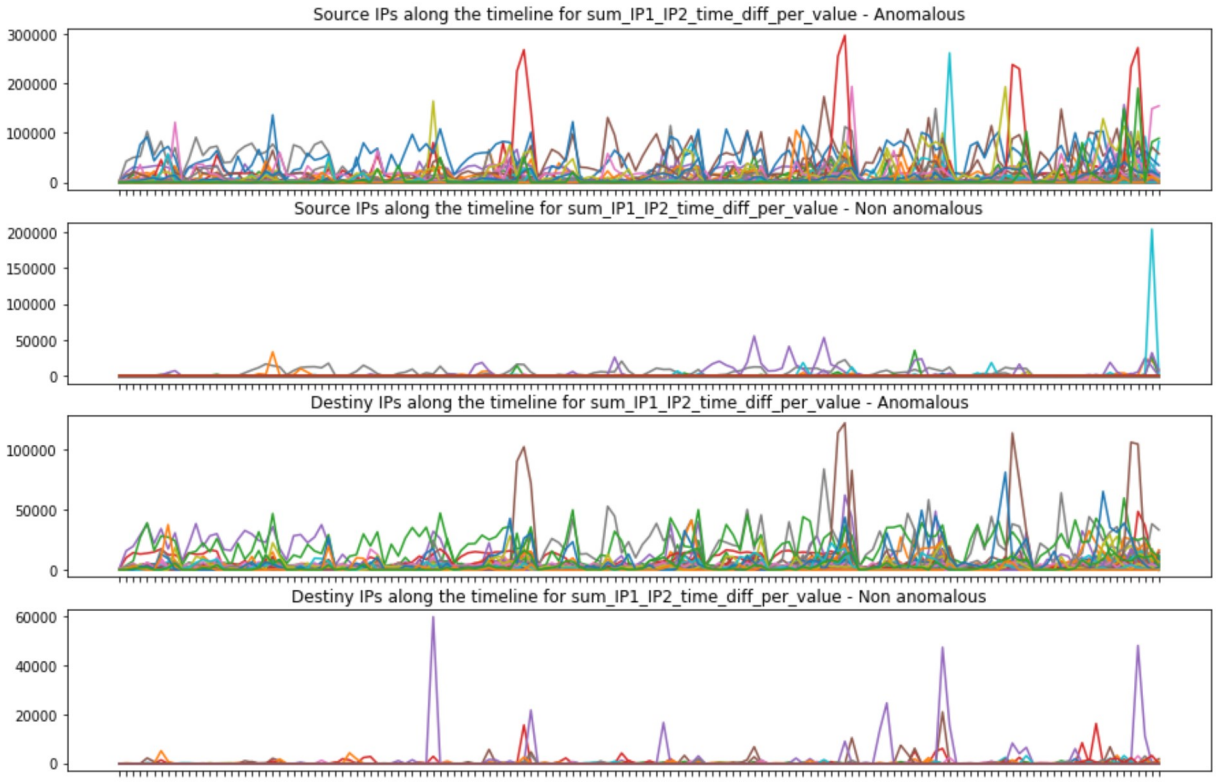

- source and destiny ips follow the same pattern on anomalous ips
- Time differences between unique ip connections are higher on anomalous ips, meaning that the same connection happens much more often on non anomalous ips

In [157]:
```
plot_all_ips(data_5T, data_5T_sample, 'std_IP1_IP2_time_diff_per_value')
```



- Similar pattern as in average
- Non anomalous have really low std, meanig that most are perodically done, whereas anomalous follow a connection pattern

In [190]:
```
plot_all_ips(data_5T, data_5T_sample, 'sum_IP1_IP2_time_diff_per_value')
```



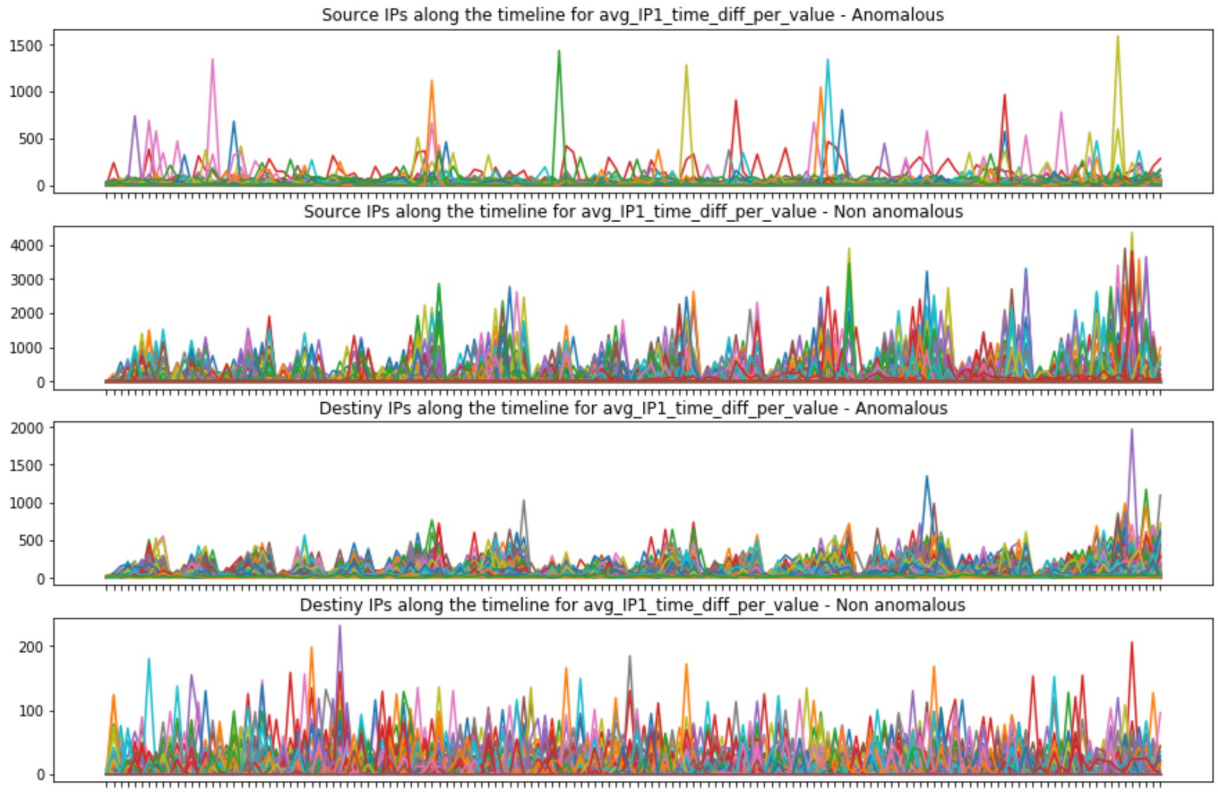

- Difference times are clearly higher in anomalous that follow the pattern

# IP1_IP2_time_diff_per_value summary

- Clear connection pattern
- Higher difference times in anomalous
- The pattern could mean that the connections are programaticaly done for anomalous observations

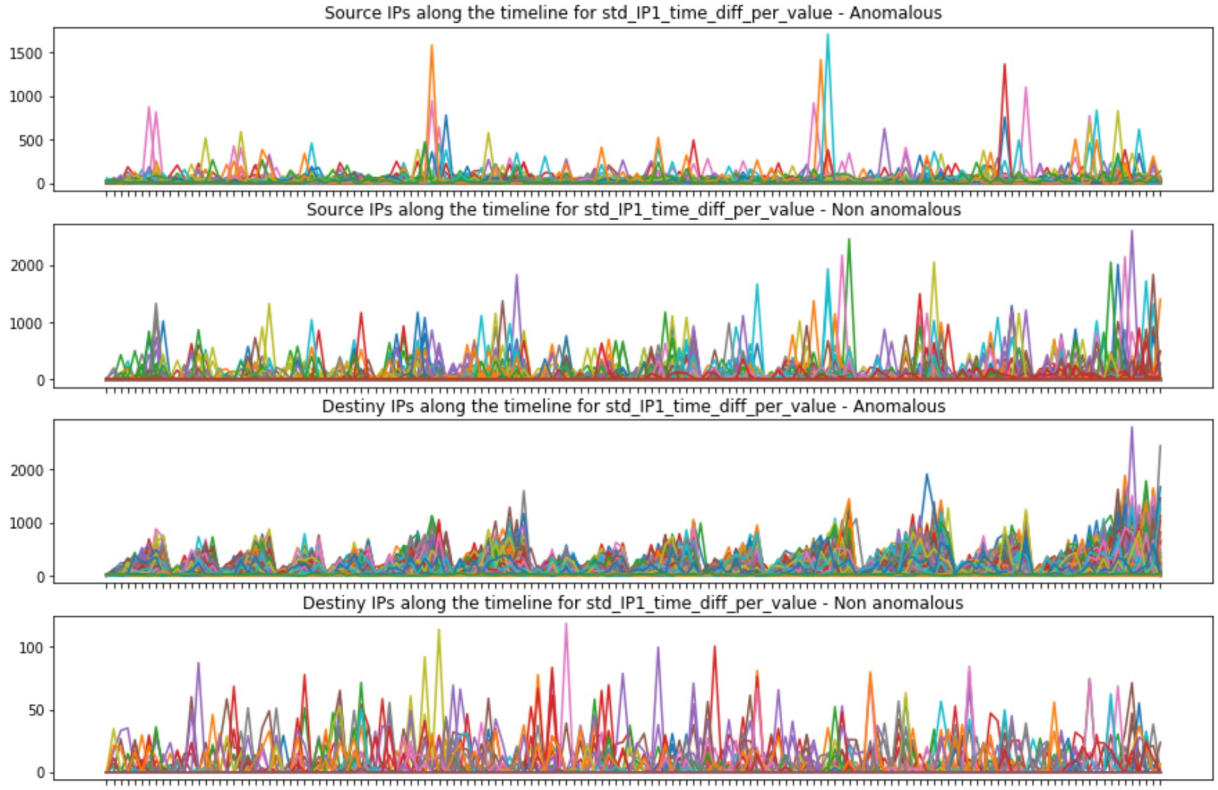
# IP1_time_diff_per_value

In [191]:
```
plot_all_ips(data_5T, data_5T_sample, 'avg_IP1_time_diff_per_value')
```



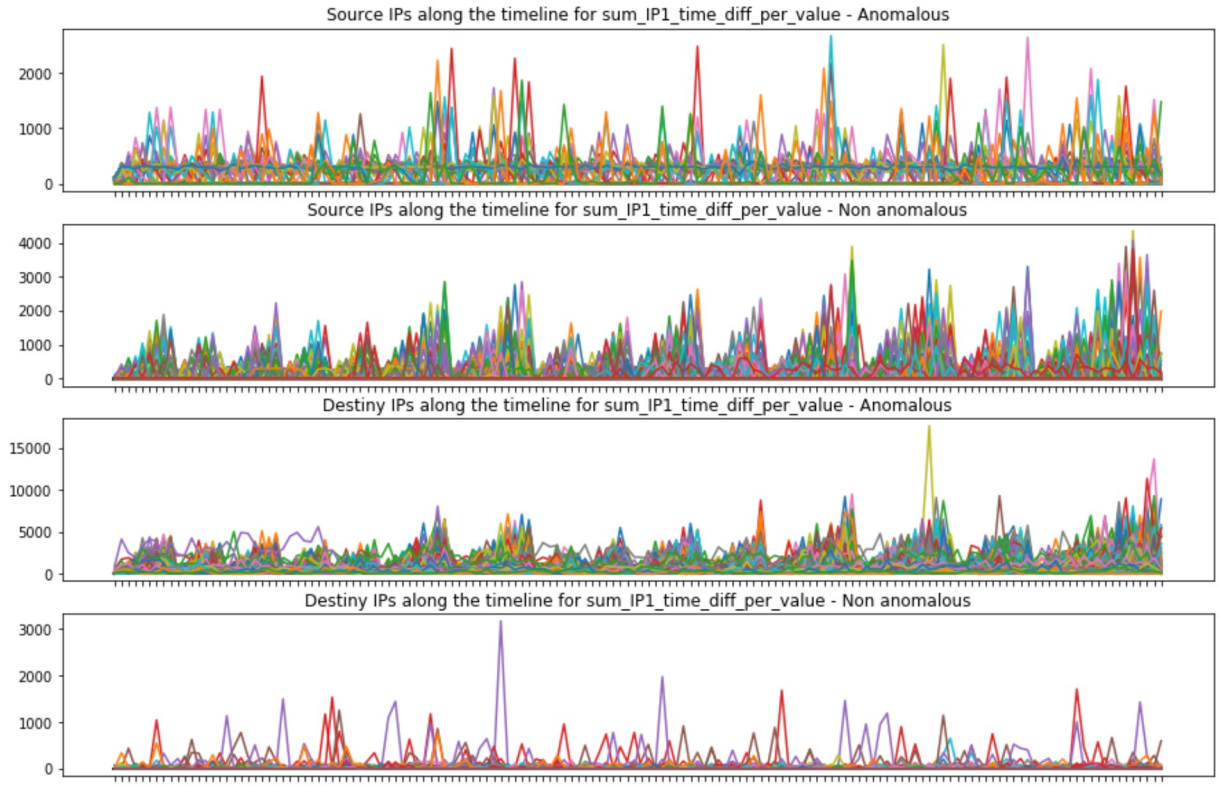

- No clear pattern on anomalous source ips, but a clear pattern on non anomalous
- Pattern on anomalous destiny ips, this could indicate that same source anomalous ip are connecting more times to different destiny ips (time difference is low), whereas the connections from source to the same destiny ip follows a pattern
- Non anomalous destiny are all over the place

In [161]:
```
plot_all_ips(data_5T, data_5T_sample, 'std_IP1_time_diff_per_value')
```




- Same conclusions as in average

In [192]:
```
plot_all_ips(data_5T, data_5T_sample, 'sum_IP1_time_diff_per_value')
```



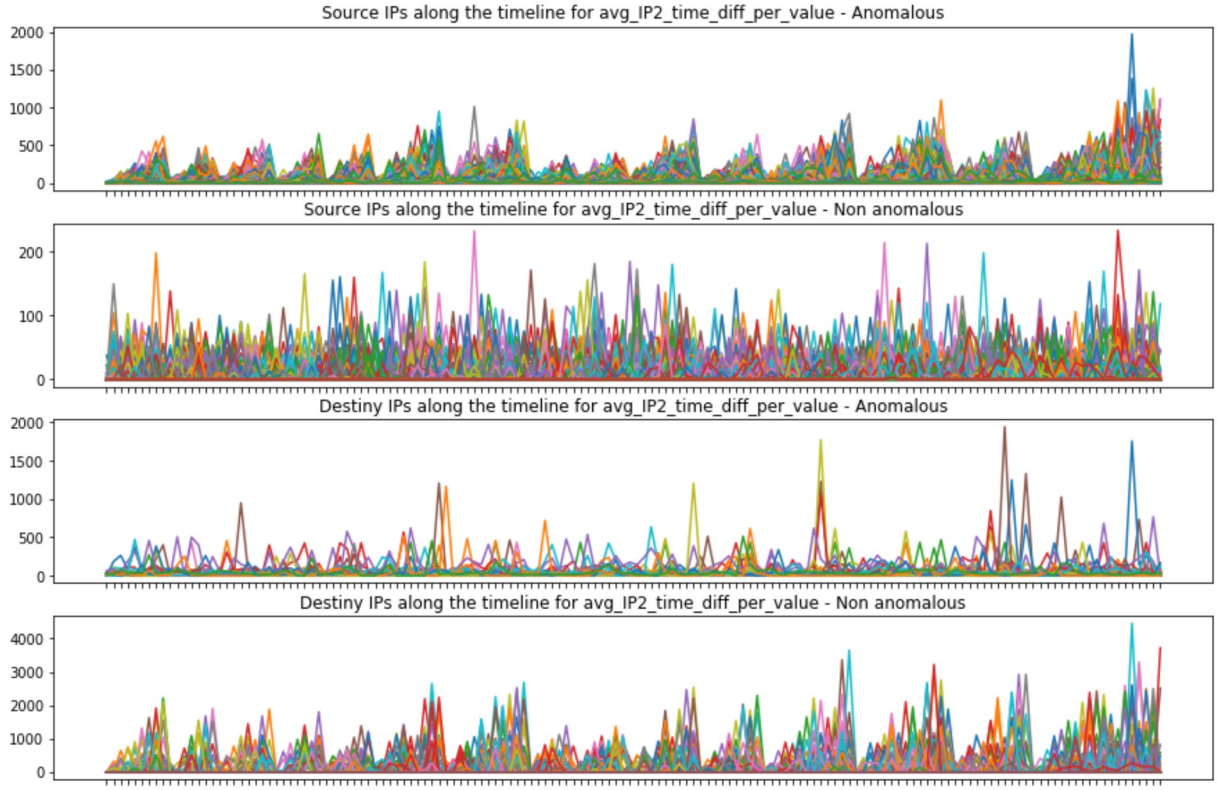
- Anomalous soures sum does not follow a clear pattern but most of the ocurrences seems to be pretty low endorsing the previous hypothesis
- Same as previous

# IP1_time_diff_per_value summary

- Anomalous source ips have really low time differences with no clear pattern as anomalous
- Source anomalous ip coulb be used to establish connections to different destiny ips
- Avg-std and sum could endorse the hypothesis
- Anomalous destiny ips follow a connection pattern from the same source ip

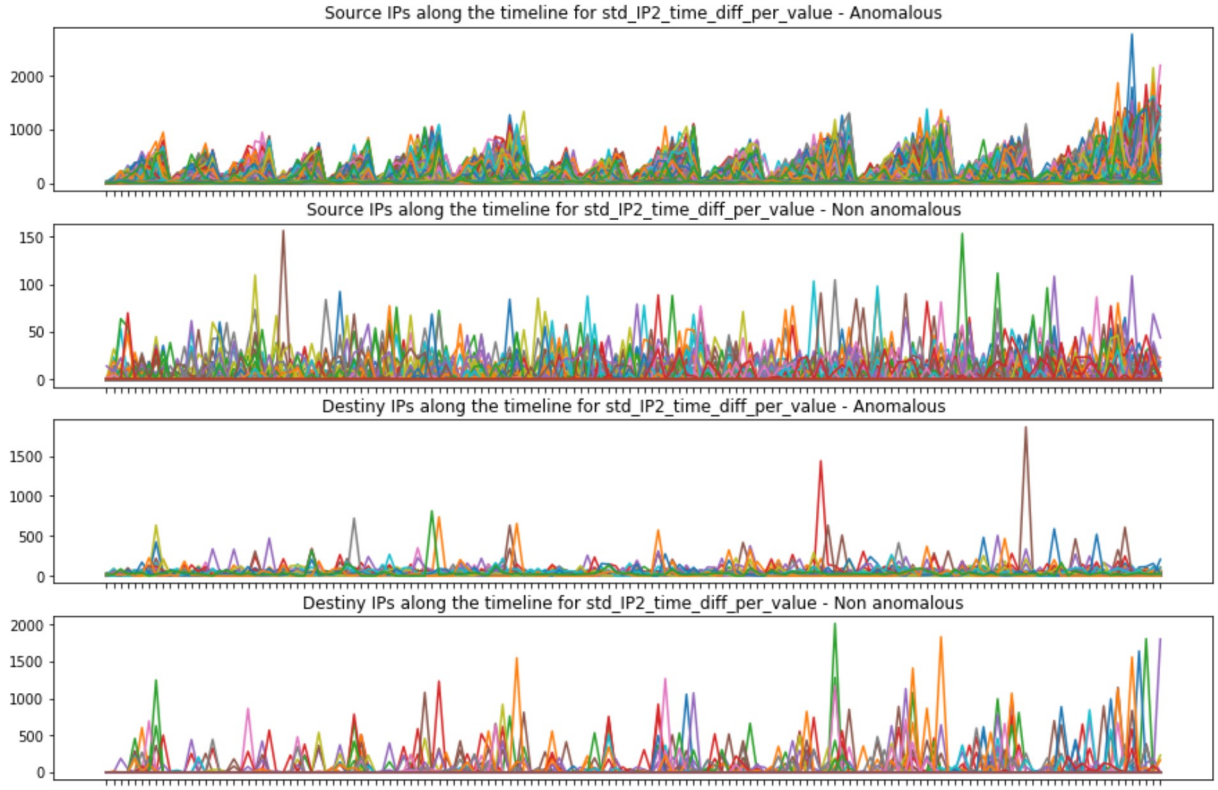
# IP2_time_diff_per_value

In [164]:
```
plot_all_ips(data_5T, data_5T_sample, 'avg_IP2_time_diff_per_value')
```



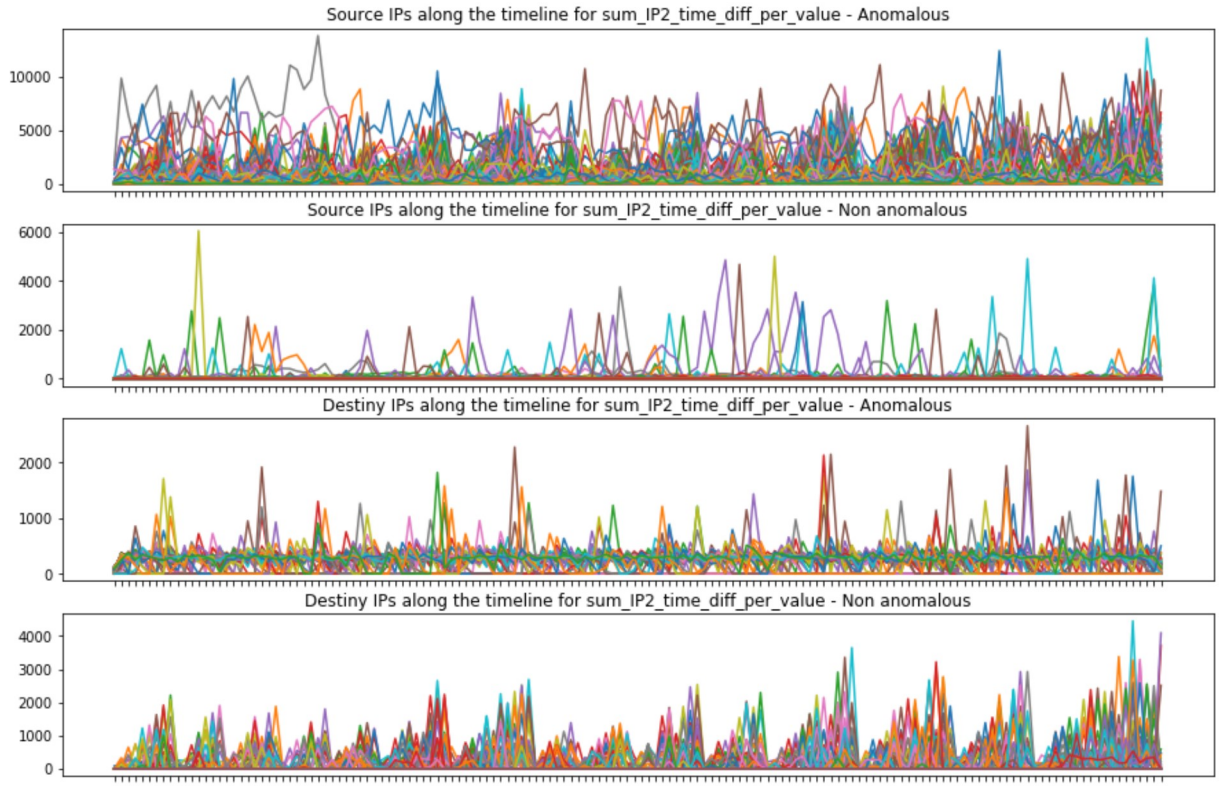

- Source anomalous ips follow a pattern similar as IP1 time diff, makes sense since the same source ips connect to different destiny ips
- Anomalous destiny ips have lower tiime differences
- Non anomalous sources are all over the place

In [165]:
```
plot_all_ips(data_5T, data_5T_sample, 'std_IP2_time_diff_per_value')
```




- Same as above and IP1 time diff

In [166]:
```
plot_all_ips(data_5T, data_5T_sample, 'sum_IP2_time_diff_per_value')
```

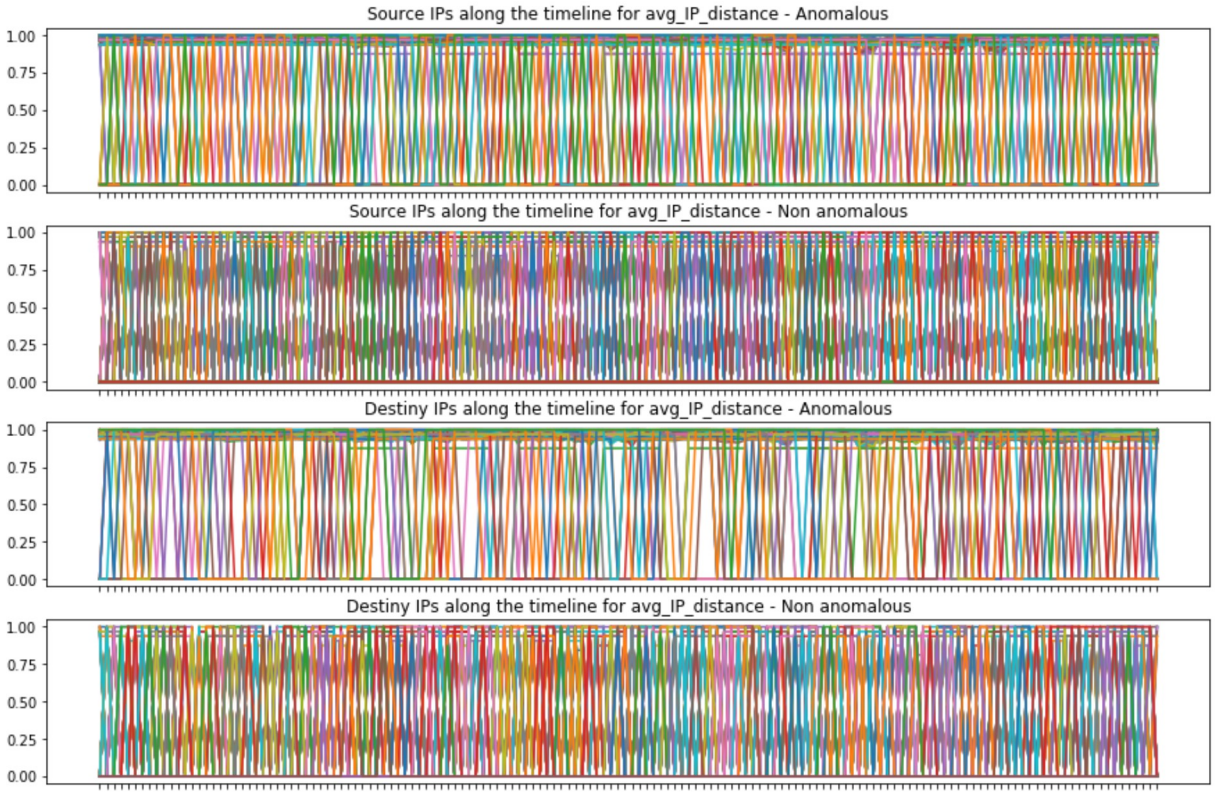


# IP2_time_diff_per_value summary

- Same hypothesis as in IP1, meaning that anomalous sources check different destiny IPs and destiny IPS are checked by different sources
- Following IP1_IP2 time diff, there could be a pattern between hosts
- Patterns between sources and destinies have switched from IP1

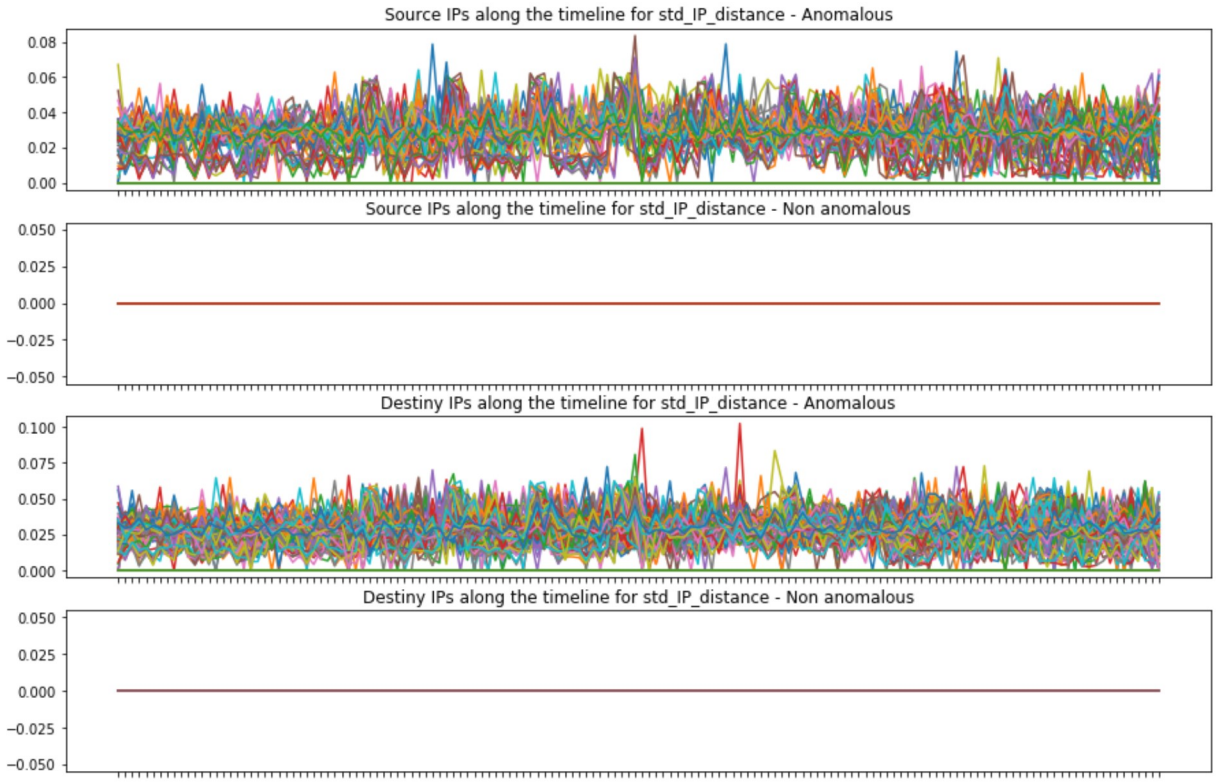
# IP_Distance

In [168]:
```
plot_all_ips(data_5T, data_5T_sample, 'avg_IP_distance')
```




- No clear pattern

In [169]:
```
plot_all_ips(data_5T, data_5T_sample, 'std_IP_distance')
```

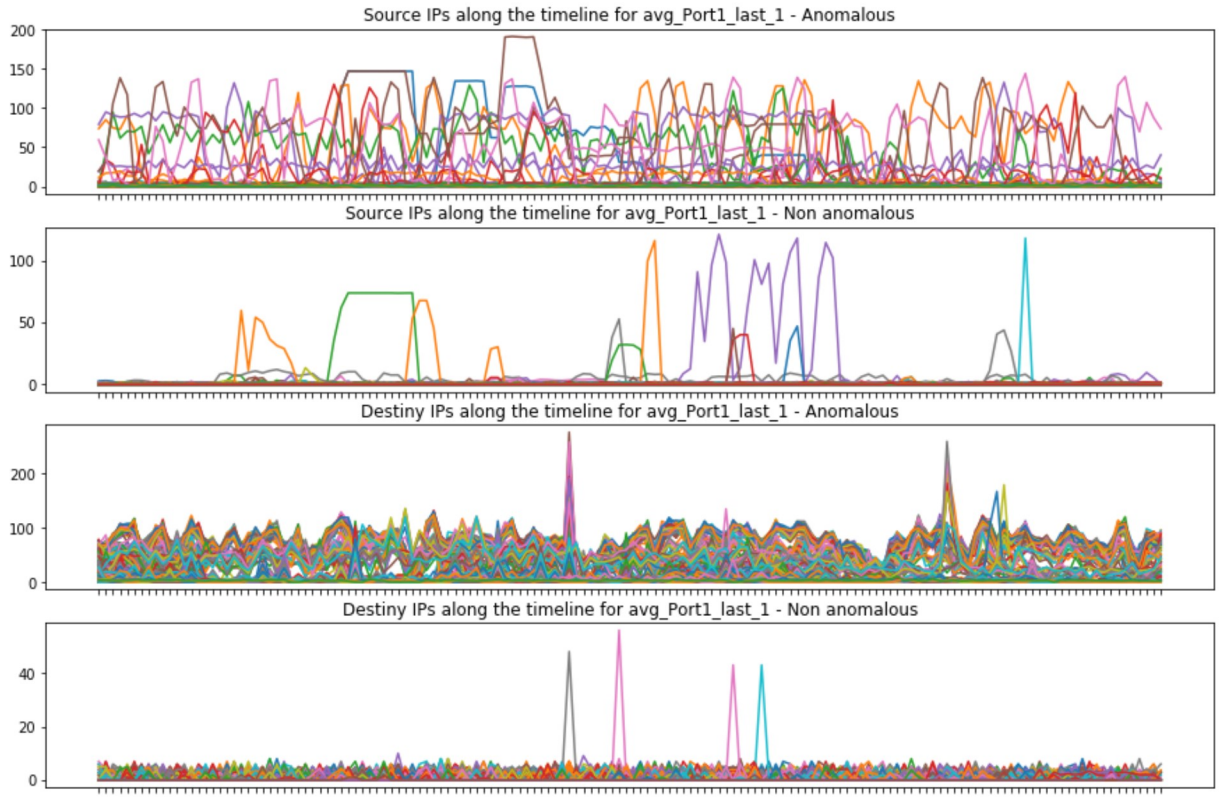



# IP_distance summary

- Anomalous have a higher std, that could indicate that anomalous connections come from closer networks than non anomalous

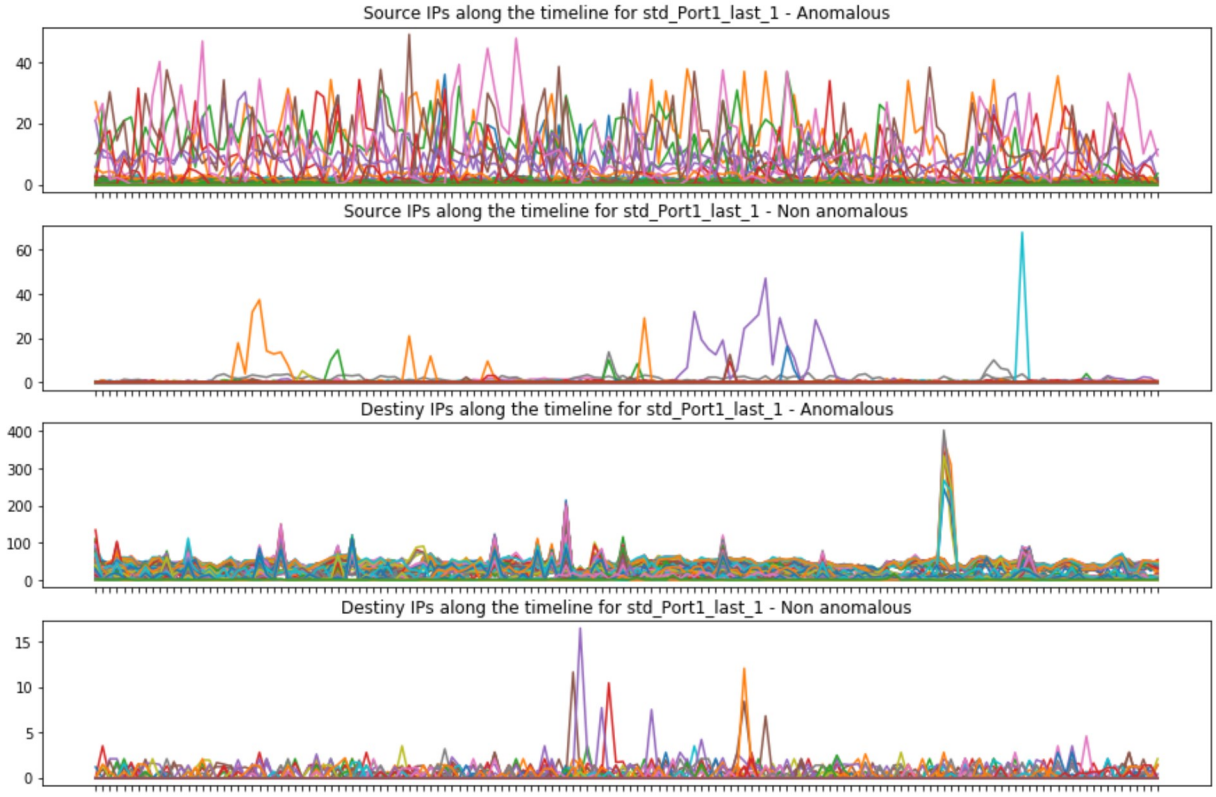
# Port1_last_1

In [172]:
```
plot_all_ips(data_5T, data_5T_sample, 'avg_Port1_last_1')
```



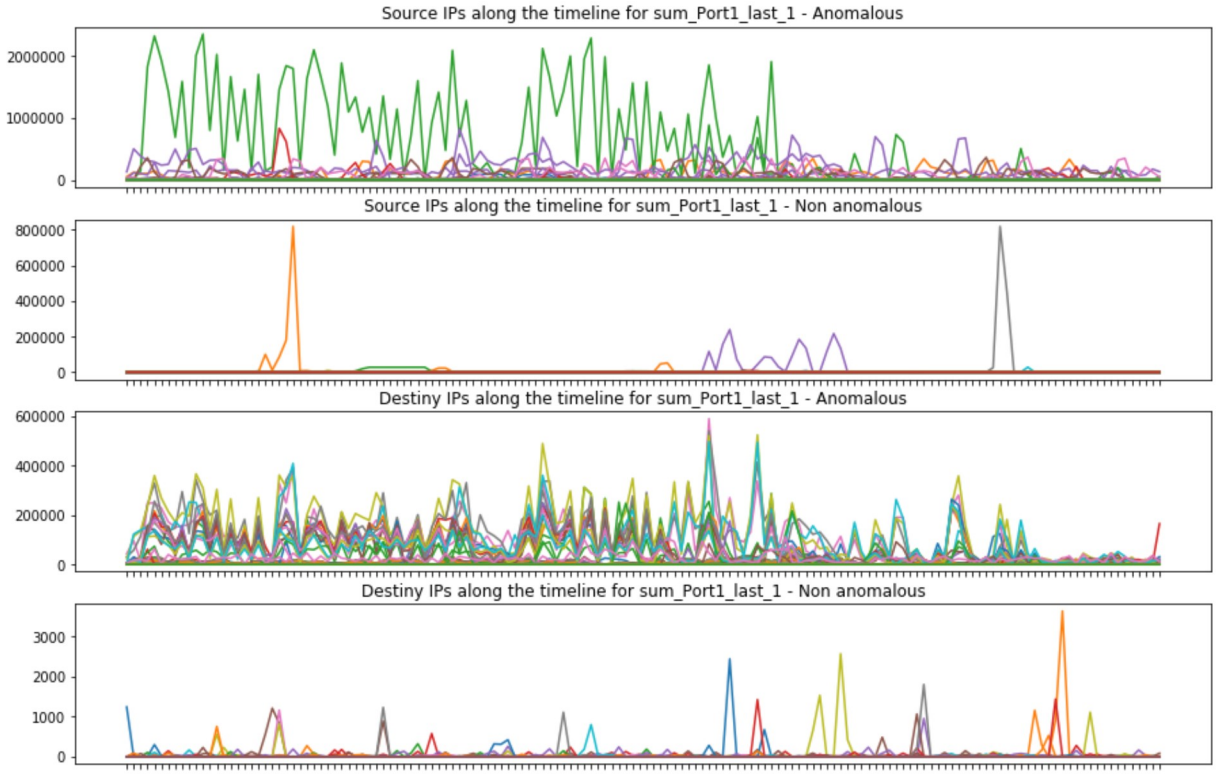

- Source anomalous have a higher use of ports thatn non anomalous, clearly
- Same four anomalous destinies, meaning that the ip that stacblished the connection to the destiny ip has used a larger amount of ports than the non anomalous counterparts

In [173]:
```
plot_all_ips(data_5T, data_5T_sample, 'std_Port1_last_1')
```




- Non anomalous tends to have lower std and lower average, meaning that the source ip uses a low number of ports during the 5 minute period

In [174]:
```
plot_all_ips(data_5T, data_5T_sample, 'sum_Port1_last_1')
```



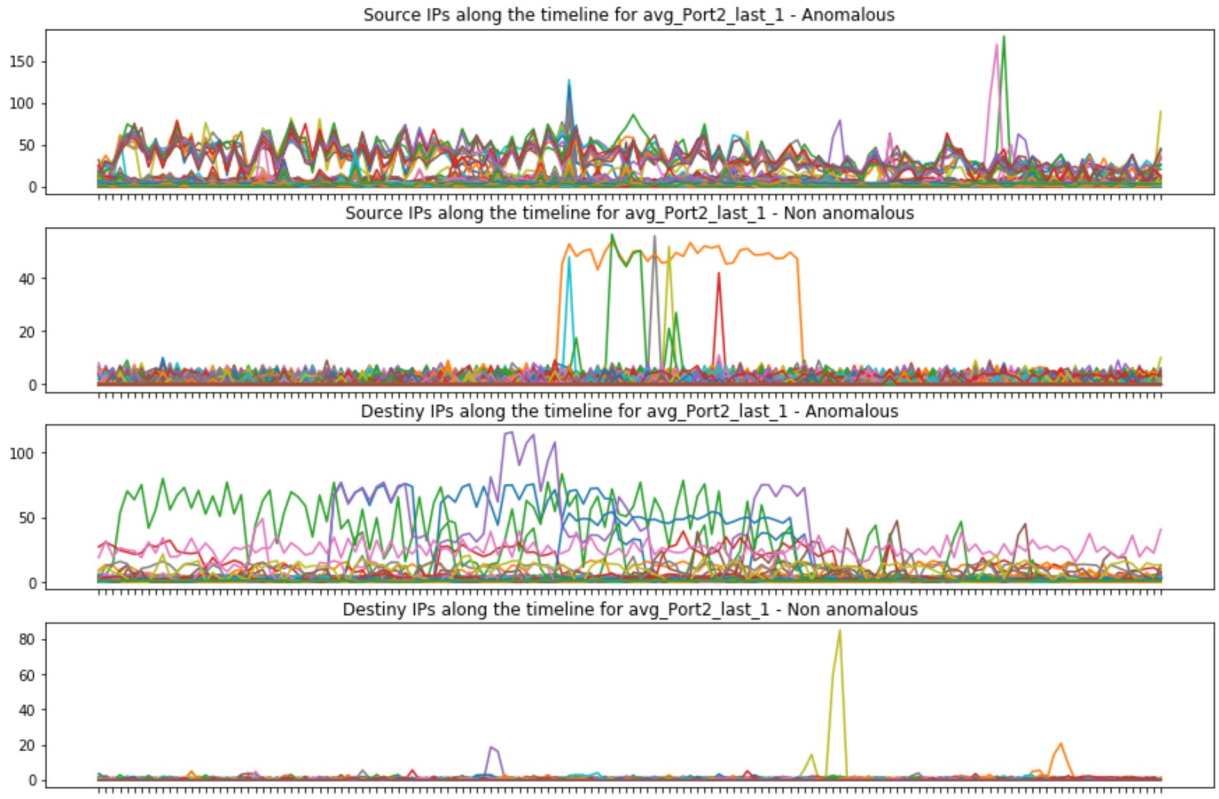
- Overall anomalous ips uses hicgher number of ports

# Port1_last_1 summary

- Anomalous IPs have tend to be connected from ips with a high port usage in the last five minutes
- Non anomalous have very little usage of ports

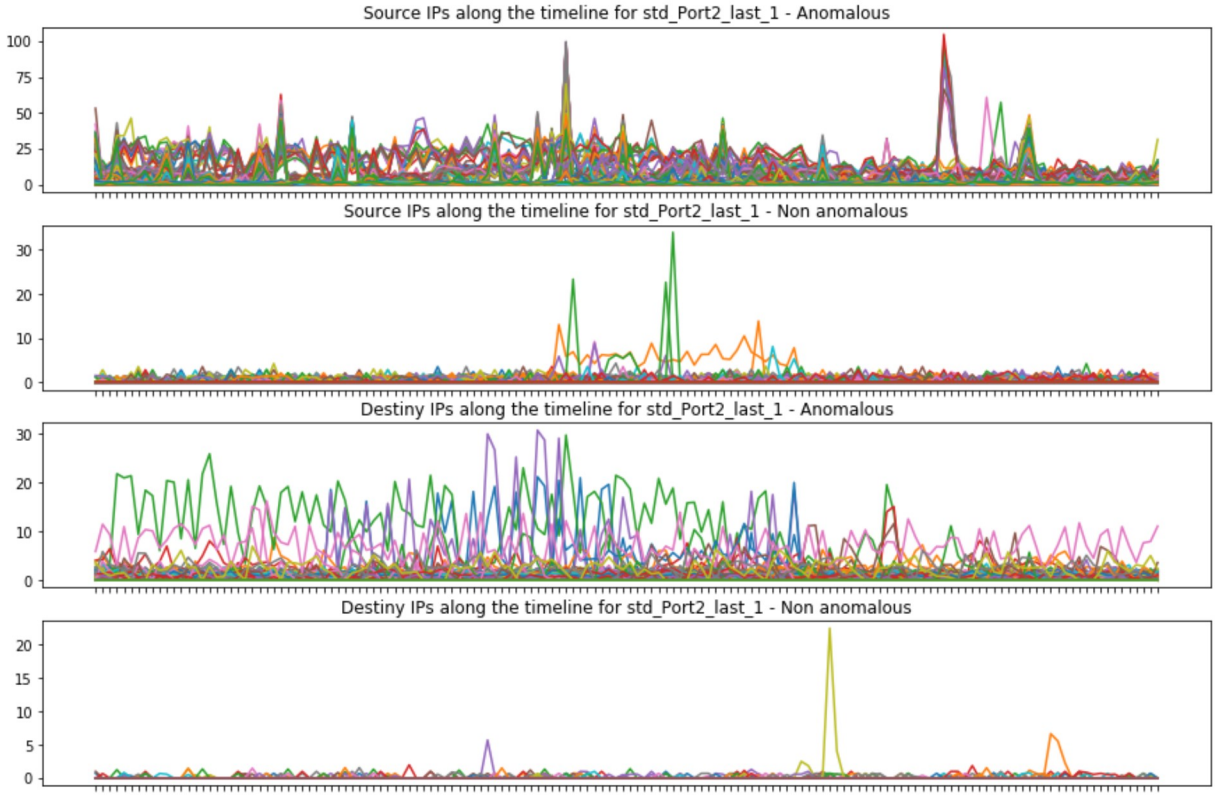
# Port2_last_1

```
In [176]: plot_all_ips(data_5T, data_5T_sample, 'avg_Port2_last_1')
```



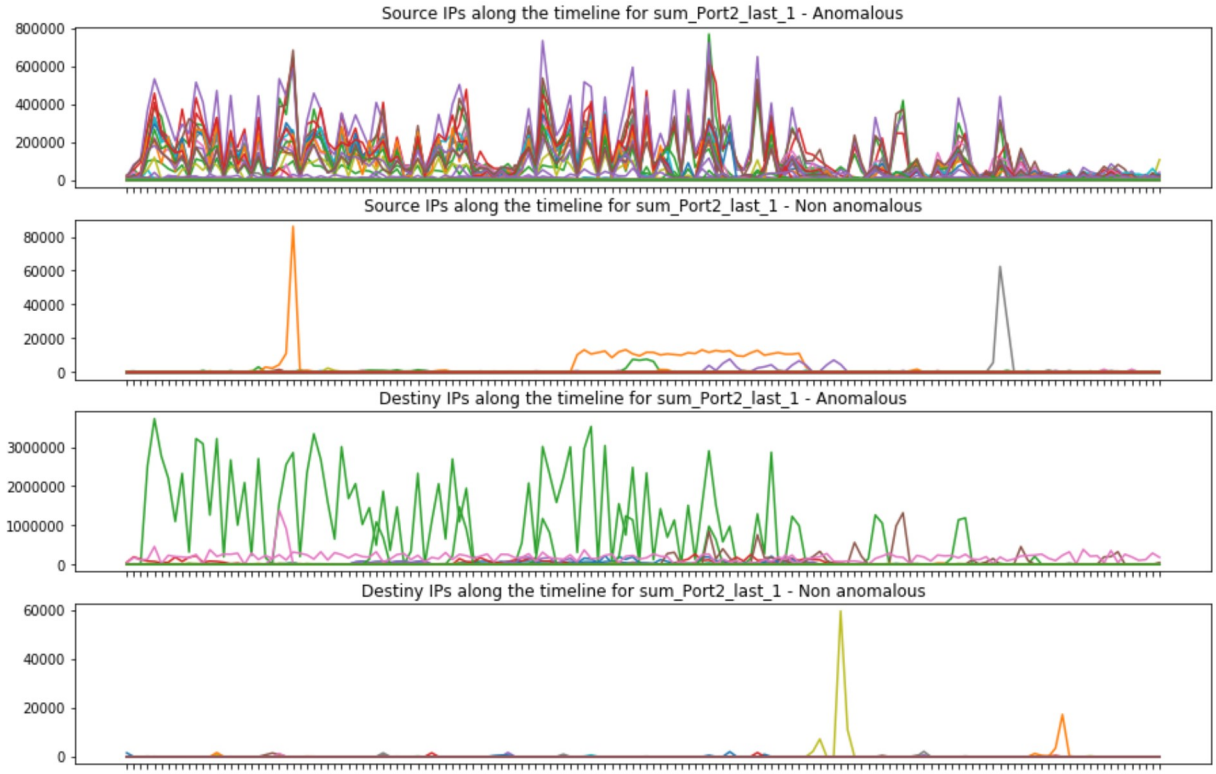

- Anomalous source connections connects to more ports than non anomalous. Meaning that the source IPs connects to different destiny ports
- Destiny ip follow the same pattern, higher port connections from the same ip. Overall destiny anomalous ips have a larger amount of connections to different ports
- Most non anomaloues have little to no port2 values

In [177]:
```
plot_all_ips(data_5T, data_5T_sample, 'std_Port2_last_1')
```




- Non anomalous observarions have low deviation

In [178]:
```
plot_all_ips(data_5T, data_5T_sample, 'sum_Port2_last_1')
```



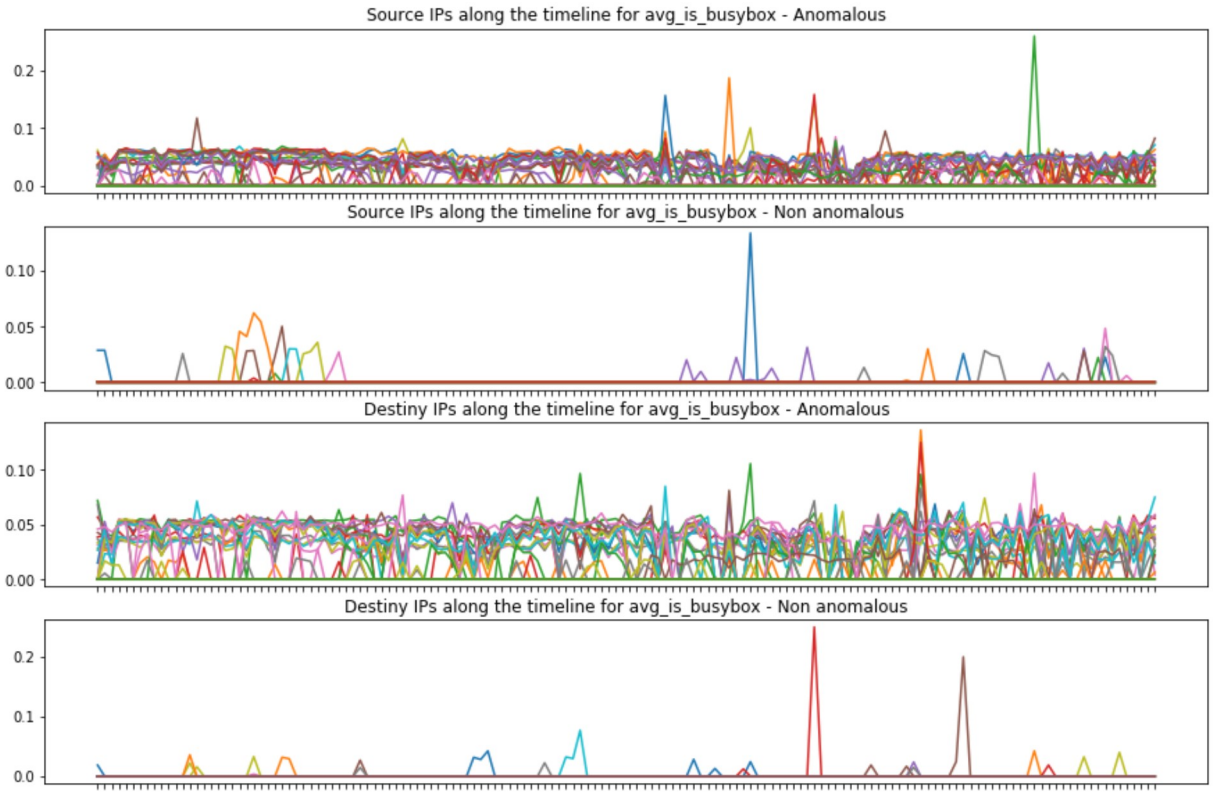
- Overall, anomalous have a higher destiny port usage than non anomalous

# Port2_last_1

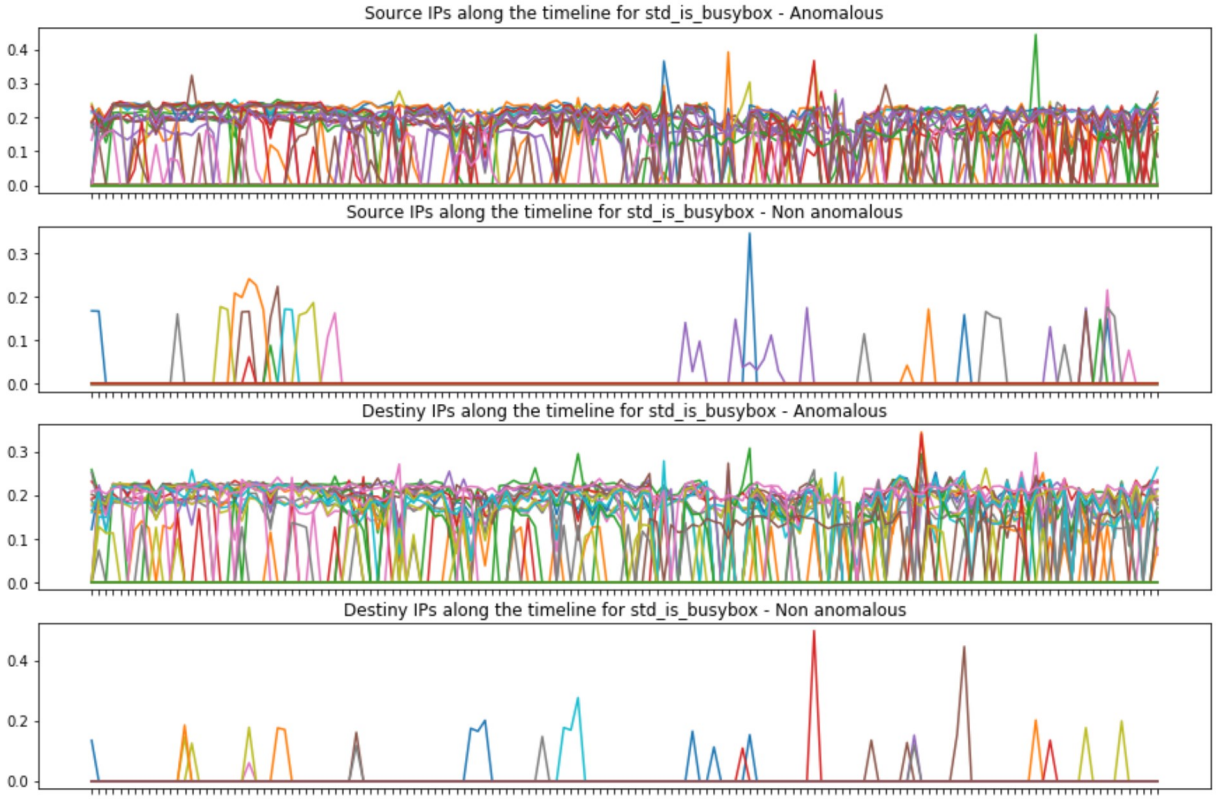
- Higher port usage on anomalous connections, this could indicate that a scanning is tacking place
- Higher port2 and port1 tends to be anomalous

# is_busybox

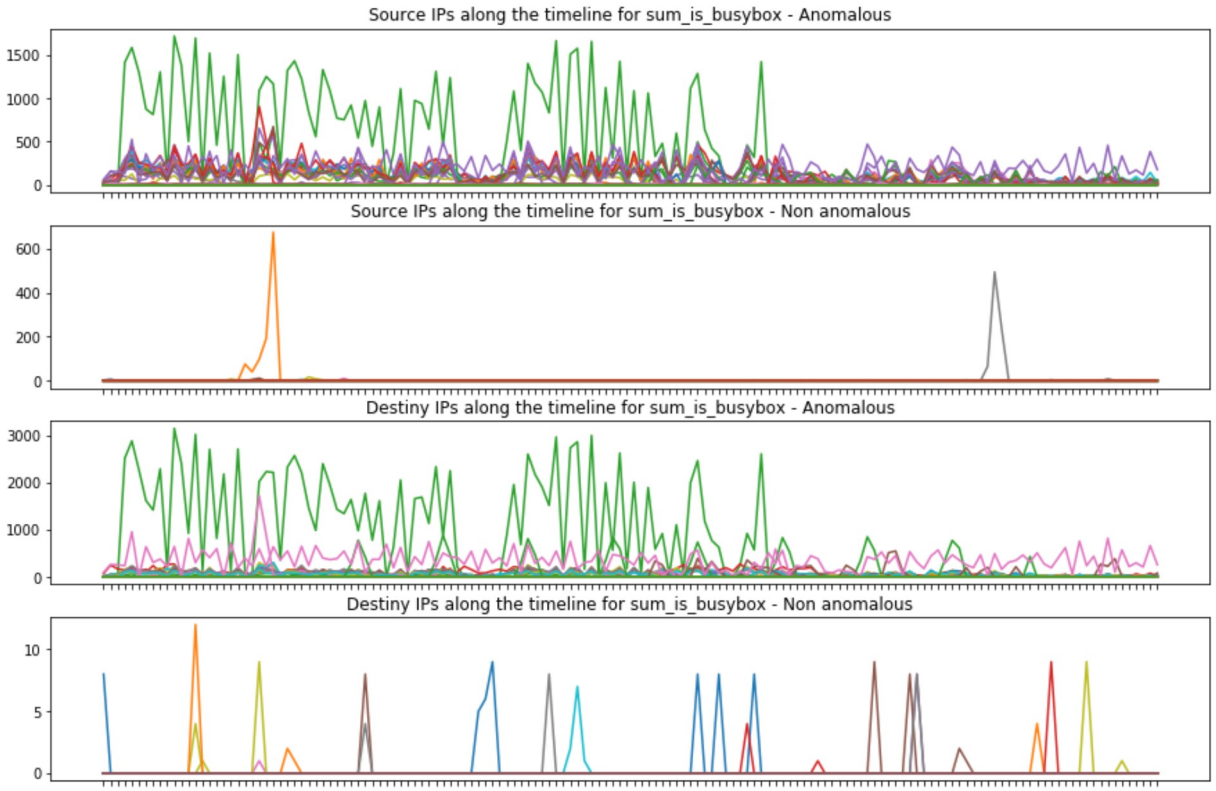
In [180]:
```
plot_all_ips(data_5T, data_5T_sample, 'avg_is_busybox')
```



In [181]:
```
plot_all_ips(data_5T, data_5T_sample, 'std_is_busybox')
```




In [182]:
```
plot_all_ips(data_5T, data_5T_sample, 'sum_is_busybox')
```

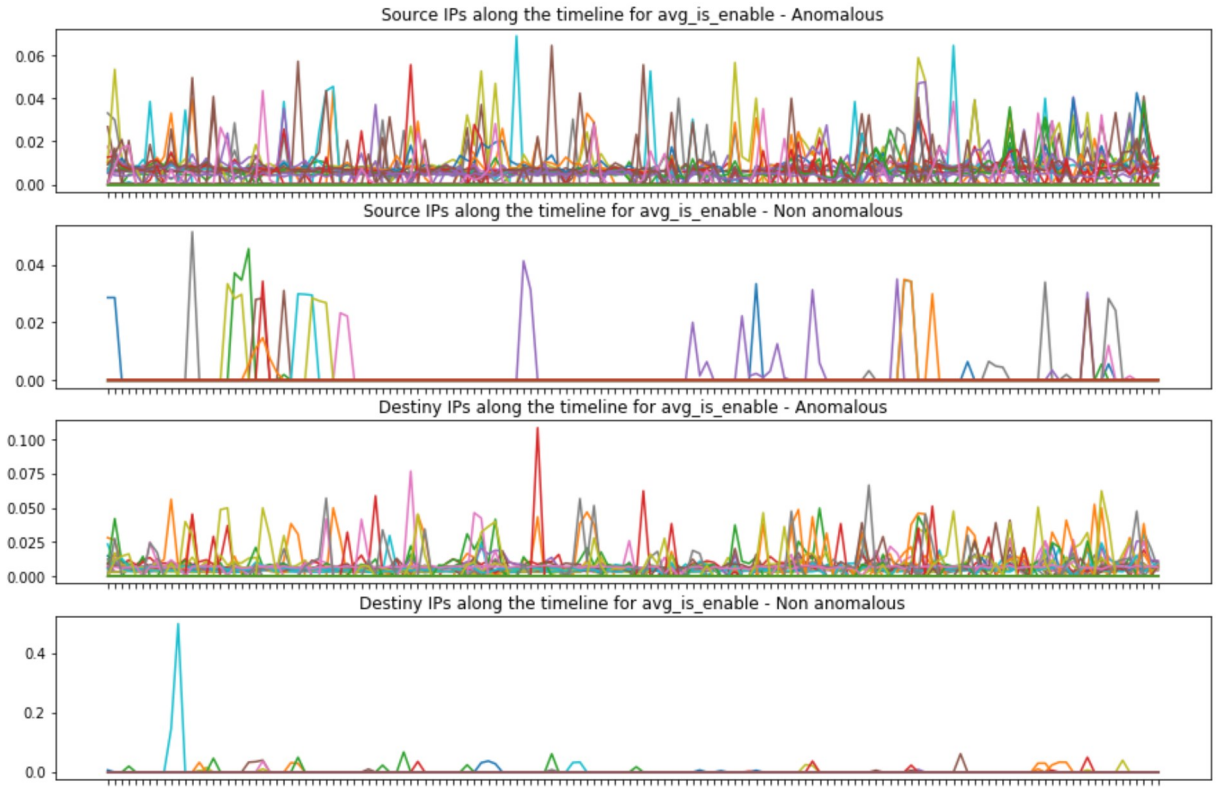


# is_busybox summary

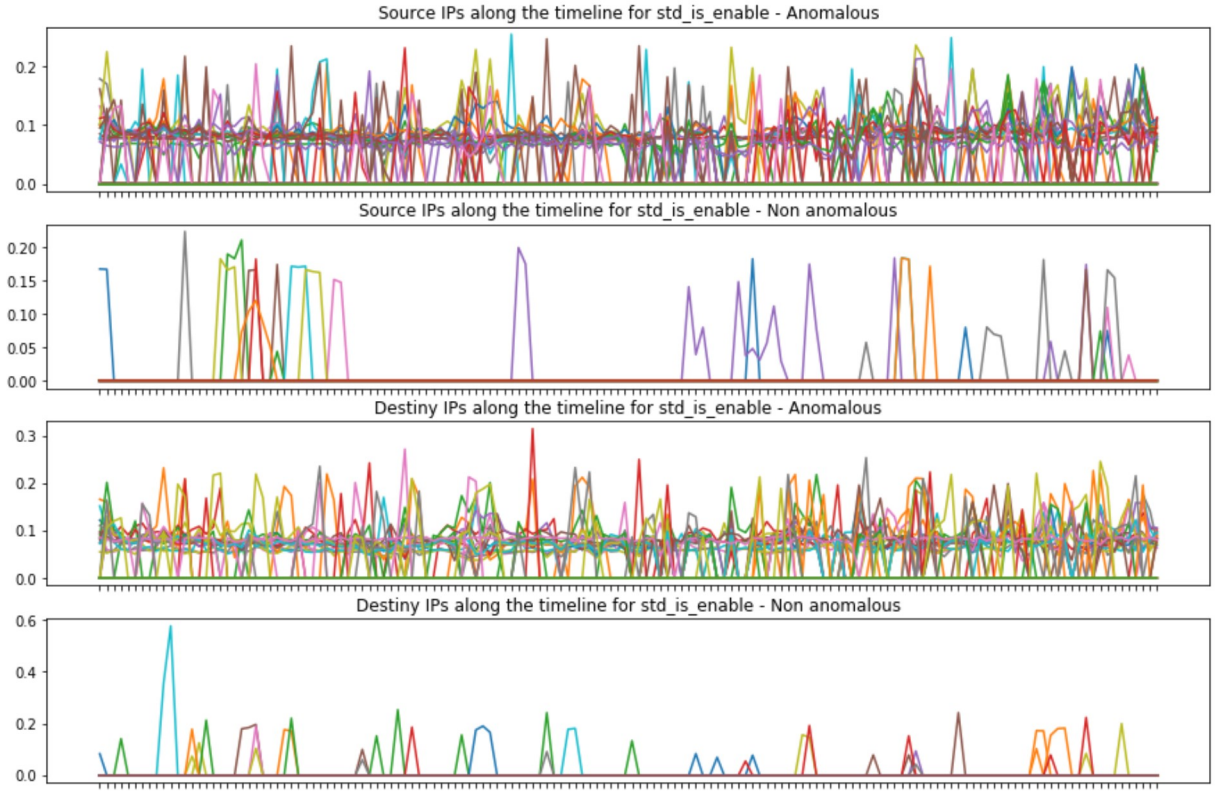
- Overall, most anomalous connections have a higher is_busybox value

# is_enable

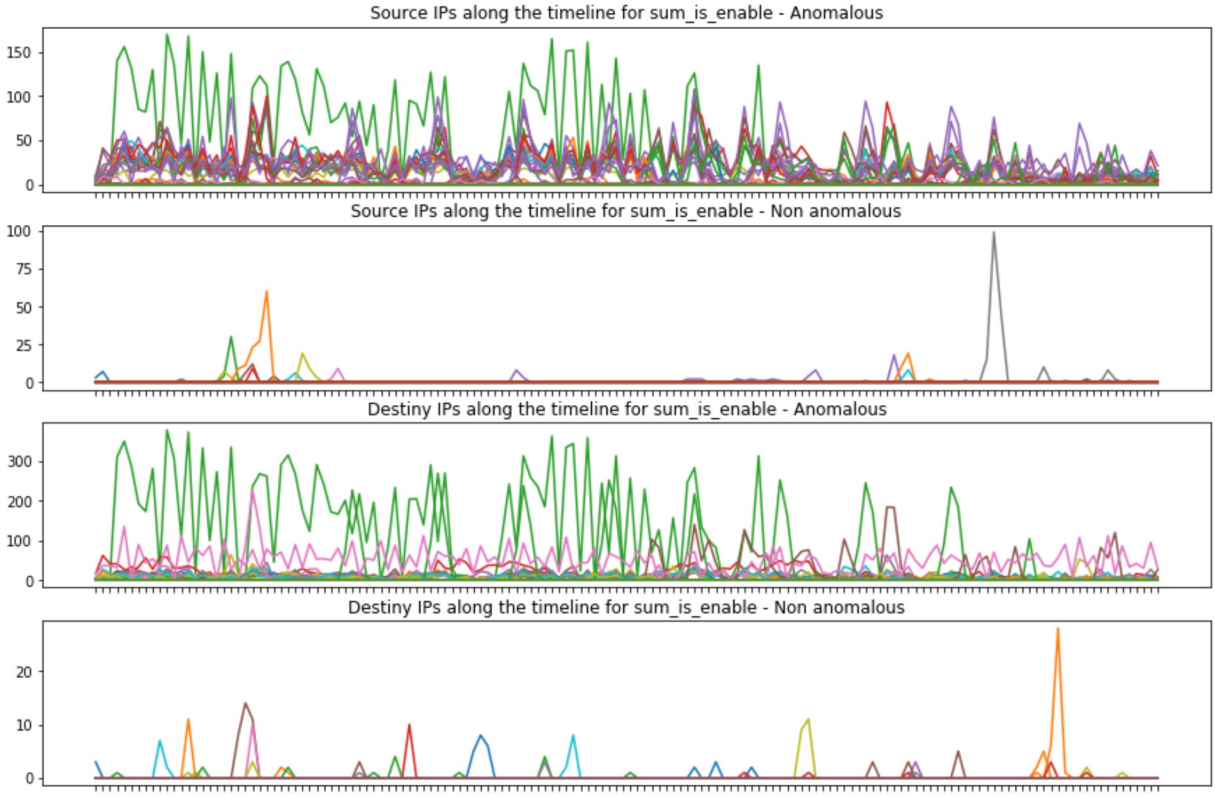
In [184]:
```
plot_all_ips(data_5T, data_5T_sample, 'avg_is_enable')
```



In [185]:
```
plot_all_ips(data_5T, data_5T_sample, 'std_is_enable')
```




In [186]:
```
plot_all_ips(data_5T, data_5T_sample, 'sum_is_enable')
```



# is_enable Summary

- Might have a higher std than non anomalous
- Some connections have a higher summ tan non anomalous, specially in source connections

is_enable and is_busybox have much more sense on source connections (connections than try to infect) than in destiny connectios, due to the source being the one sending the payload

In [ ]: