

Live 2025-11-05

November 5, 2025

## 1 2025-11-05 Live INTEUM War Stories continued

```
[1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

```
[2]: filenames = [ "../Labs/Data/4Stu_2022_1117_1219 DataWithToolInfo.csv", "../Labs/
↳Data/4Stu_2023_0118_0127 DataWithToolInfo.csv" ]

X = pd.DataFrame()

for f in filenames:
    Xn = pd.read_csv(f,
                     sep=";",
                     parse_dates = ["timestamp_skv"],
                     dayfirst = True,
                     encoding = None,
                     encoding_errors = 'replace')
    X = pd.concat([X,Xn])
```

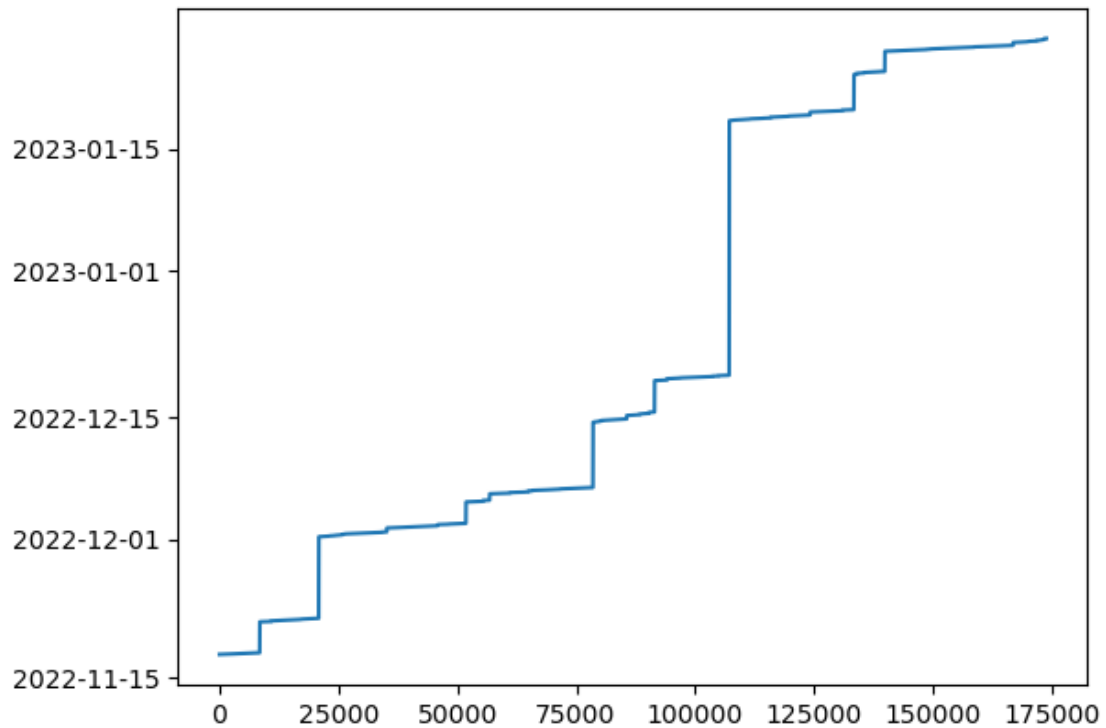
```
[3]: X.index = range(0,X.shape[0])
```

```
[4]: X.shape
```

```
[4]: (173863, 165)
```

```
[5]: plt.plot(X.index,X.timestamp_skv)
```

```
[5]: [<matplotlib.lines.Line2D at 0x7f8734784c40>]
```



[6]: *# Write code to make code :-)*

```
print("[")
for c in X.columns:
    print(f"{'{c}', "})
print("]")
```

```
[
'timestamp_skv',
'timediff_skv',
'timestamp_sensor',
'timediff_sensor',
'SynchroDiff',
'ST_CH1_P_F0',
'ST_CH1_P_F1',
'ST_CH1_P_F2',
'ST_CH1_P_F3',
'ST_CH1_P_F4',
'ST_CH1_P_F5',
'ST_CH1_P_F6',
'ST_CH1_P_F7',
'ST_CH1_P_F8',
'ST_CH1_P_F9',
'ST_CH1_P_F10',
```

'ST\_CH1\_P\_F11',  
 'ST\_CH1\_P\_Time',  
 'ST\_CH1\_P0\_F0',  
 'ST\_CH1\_P0\_F1',  
 'ST\_CH1\_P0\_F2',  
 'ST\_CH1\_P0\_F3',  
 'ST\_CH1\_P0\_F4',  
 'ST\_CH1\_P0\_F5',  
 'ST\_CH1\_P0\_F6',  
 'ST\_CH1\_P0\_F7',  
 'ST\_CH1\_P0\_F8',  
 'ST\_CH1\_P0\_F9',  
 'ST\_CH1\_P0\_F10',  
 'ST\_CH1\_P0\_F11',  
 'ST\_CH1\_P0\_Time',  
 'ST\_CH1\_P1\_F0',  
 'ST\_CH1\_P1\_F1',  
 'ST\_CH1\_P1\_F2',  
 'ST\_CH1\_P1\_F3',  
 'ST\_CH1\_P1\_F4',  
 'ST\_CH1\_P1\_F5',  
 'ST\_CH1\_P1\_F6',  
 'ST\_CH1\_P1\_F7',  
 'ST\_CH1\_P1\_F8',  
 'ST\_CH1\_P1\_F9',  
 'ST\_CH1\_P1\_F10',  
 'ST\_CH1\_P1\_F11',  
 'ST\_CH1\_P1\_Time',  
 'ST\_CH1\_P2\_F0',  
 'ST\_CH1\_P2\_F1',  
 'ST\_CH1\_P2\_F2',  
 'ST\_CH1\_P2\_F3',  
 'ST\_CH1\_P2\_F4',  
 'ST\_CH1\_P2\_F5',  
 'ST\_CH1\_P2\_F6',  
 'ST\_CH1\_P2\_F7',  
 'ST\_CH1\_P2\_F8',  
 'ST\_CH1\_P2\_F9',  
 'ST\_CH1\_P2\_F10',  
 'ST\_CH1\_P2\_F11',  
 'ST\_CH1\_P2\_Time',  
 'ST\_CH2\_P\_F0',  
 'ST\_CH2\_P\_F1',  
 'ST\_CH2\_P\_F2',  
 'ST\_CH2\_P\_F3',  
 'ST\_CH2\_P\_F4',  
 'ST\_CH2\_P\_F5',  
 'ST\_CH2\_P\_F6',

'ST\_CH2\_P\_F7',  
'ST\_CH2\_P\_F8',  
'ST\_CH2\_P\_F9',  
'ST\_CH2\_P\_F10',  
'ST\_CH2\_P\_F11',  
'ST\_CH2\_P\_Time',  
'ST\_CH2\_P0\_F0',  
'ST\_CH2\_P0\_F1',  
'ST\_CH2\_P0\_F2',  
'ST\_CH2\_P0\_F3',  
'ST\_CH2\_P0\_F4',  
'ST\_CH2\_P0\_F5',  
'ST\_CH2\_P0\_F6',  
'ST\_CH2\_P0\_F7',  
'ST\_CH2\_P0\_F8',  
'ST\_CH2\_P0\_F9',  
'ST\_CH2\_P0\_F10',  
'ST\_CH2\_P0\_F11',  
'ST\_CH2\_P0\_Time',  
'ST\_CH2\_P1\_F0',  
'ST\_CH2\_P1\_F1',  
'ST\_CH2\_P1\_F2',  
'ST\_CH2\_P1\_F3',  
'ST\_CH2\_P1\_F4',  
'ST\_CH2\_P1\_F5',  
'ST\_CH2\_P1\_F6',  
'ST\_CH2\_P1\_F7',  
'ST\_CH2\_P1\_F8',  
'ST\_CH2\_P1\_F9',  
'ST\_CH2\_P1\_F10',  
'ST\_CH2\_P1\_F11',  
'ST\_CH2\_P1\_Time',  
'ST\_CH2\_P2\_F0',  
'ST\_CH2\_P2\_F1',  
'ST\_CH2\_P2\_F2',  
'ST\_CH2\_P2\_F3',  
'ST\_CH2\_P2\_F4',  
'ST\_CH2\_P2\_F5',  
'ST\_CH2\_P2\_F6',  
'ST\_CH2\_P2\_F7',  
'ST\_CH2\_P2\_F8',  
'ST\_CH2\_P2\_F9',  
'ST\_CH2\_P2\_F10',  
'ST\_CH2\_P2\_F11',  
'ST\_CH2\_P2\_Time',  
'4.713 SKV#1',  
'0.604 min SKV#1',  
'0.604 max SKV#1',

'0.74 min SKV#1',  
'0.74 max SKV#1',  
'4,96 SKV#1',  
'4,40 SKV#1',  
'5,36 SKV#1',  
'5,92 SKV#1',  
'Sym aussen SKV#1',  
'Sym Einstich SKV#1',  
'Licht-Einstich frei SKV#1',  
'4.713 SKV#2',  
'0.604 min SKV#2',  
'0.604 max SKV#2',  
'0.74 min SKV#2',  
'0.74 max SKV#2',  
'4,96 SKV#2',  
'4,40 SKV#2',  
'5,36 SKV#2',  
'5,92 SKV#2',  
'Sym Einstich SKV#2',  
'Sym aussen SKV#2',  
'Licht-Einstich frei SKV#2',  
'4.713 SKV#3',  
'0.604 min SKV#3',  
'0.604 max SKV#3',  
'0.74 min SKV#3',  
'0.74 max SKV#3',  
'4,96 SKV#3',  
'4,40 SKV#3',  
'5,36 SKV#3',  
'5,92 SKV#3',  
'Sym Einstich SKV#3',  
'Sym aussen SKV#3',  
'Licht-Einstich frei SKV#3',  
'4.713 SKV#4',  
'0.604 min SKV#4',  
'0.604 max SKV#4',  
'0.74 min SKV#4',  
'0.74 max SKV#4',  
'4,96 SKV#4',  
'4,40 SKV#4',  
'5,36 SKV#4',  
'5,92 SKV#4',  
'Sym Einstich SKV#4',  
'Sym aussen SKV#4',  
'Licht-Einstich frei SKV#4',  
'timestamp',  
'Beacon INT0028 - BME280\_RH',  
'timestamp.1',

```

'Beacon INT0028 - BMP280_PRESS',
'timestamp.2',
'Beacon INT0028 - BMP280_TEMP',
'pieces',
'tool io',
]

```

```

[7]: interestingColumns =[
    # 'timestamp_skv',
    # 'timediff_skv',
    # 'timestamp_sensor',
    # 'timediff_sensor',
    # 'SynchroDiff',
    'ST_CH1_P_F0',
    'ST_CH1_P_F1',
    'ST_CH1_P_F2',
    'ST_CH1_P_F3',
    'ST_CH1_P_F4',
    'ST_CH1_P_F5',
    'ST_CH1_P_F6',
    'ST_CH1_P_F7',
    'ST_CH1_P_F8',
    'ST_CH1_P_F9',
    'ST_CH1_P_F10',
    'ST_CH1_P_F11',
    # 'ST_CH1_P_Time',
    'ST_CH1_P0_F0',
    'ST_CH1_P0_F1',
    'ST_CH1_P0_F2',
    'ST_CH1_P0_F3',
    'ST_CH1_P0_F4',
    'ST_CH1_P0_F5',
    'ST_CH1_P0_F6',
    'ST_CH1_P0_F7',
    'ST_CH1_P0_F8',
    'ST_CH1_P0_F9',
    'ST_CH1_P0_F10',
    'ST_CH1_P0_F11',
    # 'ST_CH1_P0_Time',
    'ST_CH1_P1_F0',
    'ST_CH1_P1_F1',
    'ST_CH1_P1_F2',
    'ST_CH1_P1_F3',
    'ST_CH1_P1_F4',
    'ST_CH1_P1_F5',
    'ST_CH1_P1_F6',
    'ST_CH1_P1_F7',

```

```

'ST_CH1_P1_F8',
'ST_CH1_P1_F9',
'ST_CH1_P1_F10',
'ST_CH1_P1_F11',
# 'ST_CH1_P1_Time',
'ST_CH1_P2_F0',
'ST_CH1_P2_F1',
'ST_CH1_P2_F2',
'ST_CH1_P2_F3',
'ST_CH1_P2_F4',
'ST_CH1_P2_F5',
'ST_CH1_P2_F6',
'ST_CH1_P2_F7',
'ST_CH1_P2_F8',
'ST_CH1_P2_F9',
'ST_CH1_P2_F10',
'ST_CH1_P2_F11',
# 'ST_CH1_P2_Time',
'ST_CH2_P_F0',
'ST_CH2_P_F1',
'ST_CH2_P_F2',
'ST_CH2_P_F3',
'ST_CH2_P_F4',
'ST_CH2_P_F5',
'ST_CH2_P_F6',
'ST_CH2_P_F7',
'ST_CH2_P_F8',
'ST_CH2_P_F9',
'ST_CH2_P_F10',
'ST_CH2_P_F11',
# 'ST_CH2_P_Time',
'ST_CH2_P0_F0',
'ST_CH2_P0_F1',
'ST_CH2_P0_F2',
'ST_CH2_P0_F3',
'ST_CH2_P0_F4',
'ST_CH2_P0_F5',
'ST_CH2_P0_F6',
'ST_CH2_P0_F7',
'ST_CH2_P0_F8',
'ST_CH2_P0_F9',
'ST_CH2_P0_F10',
'ST_CH2_P0_F11',
# 'ST_CH2_P0_Time',
'ST_CH2_P1_F0',
'ST_CH2_P1_F1',
'ST_CH2_P1_F2',

```

```
'ST_CH2_P1_F3',
'ST_CH2_P1_F4',
'ST_CH2_P1_F5',
'ST_CH2_P1_F6',
'ST_CH2_P1_F7',
'ST_CH2_P1_F8',
'ST_CH2_P1_F9',
'ST_CH2_P1_F10',
'ST_CH2_P1_F11',
# 'ST_CH2_P1_Time',
'ST_CH2_P2_F0',
'ST_CH2_P2_F1',
'ST_CH2_P2_F2',
'ST_CH2_P2_F3',
'ST_CH2_P2_F4',
'ST_CH2_P2_F5',
'ST_CH2_P2_F6',
'ST_CH2_P2_F7',
'ST_CH2_P2_F8',
'ST_CH2_P2_F9',
'ST_CH2_P2_F10',
'ST_CH2_P2_F11',
# 'ST_CH2_P2_Time',
'4.713 SKV#1',
'0.604 min SKV#1',
'0.604 max SKV#1',
'0.74 min SKV#1',
'0.74 max SKV#1',
'4,96 SKV#1',
'4,40 SKV#1',
'5,36 SKV#1',
'5,92 SKV#1',
'Sym aussen SKV#1',
'Sym Einstich SKV#1',
'Licht-Einstich frei SKV#1',
'4.713 SKV#2',
'0.604 min SKV#2',
'0.604 max SKV#2',
'0.74 min SKV#2',
'0.74 max SKV#2',
'4,96 SKV#2',
'4,40 SKV#2',
'5,36 SKV#2',
'5,92 SKV#2',
'Sym Einstich SKV#2',
'Sym aussen SKV#2',
'Licht-Einstich frei SKV#2',
```



```

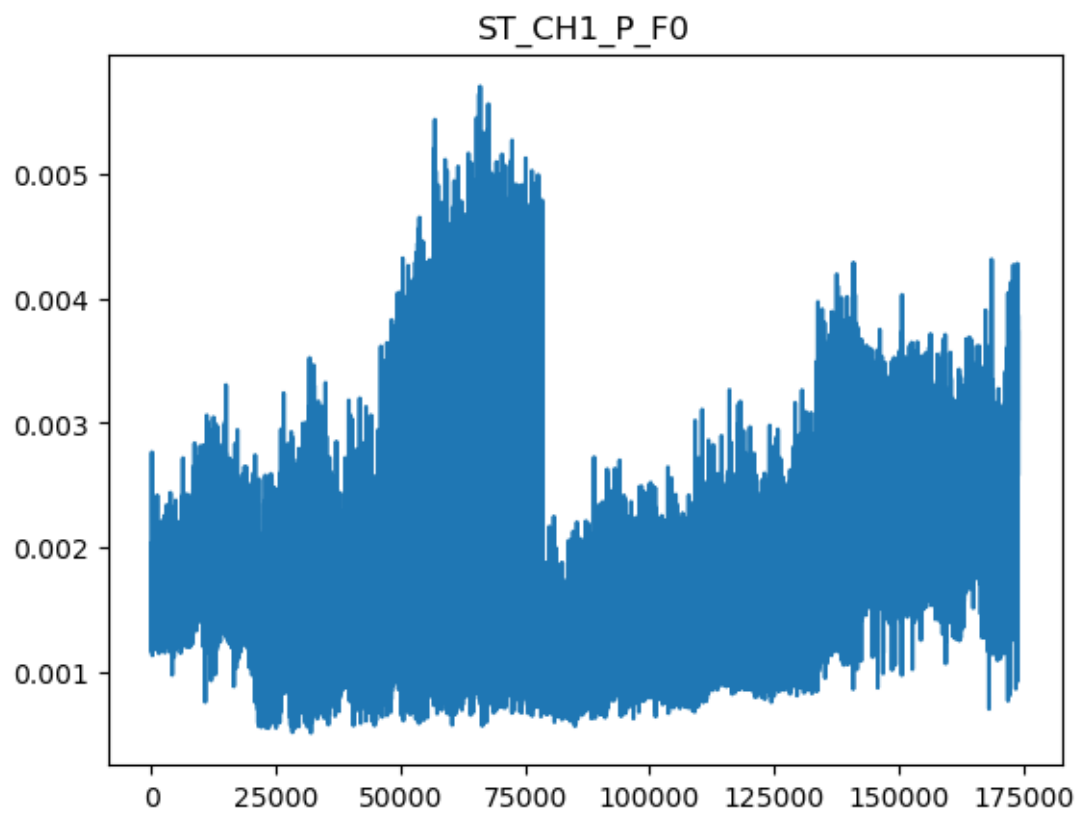
'4.713 SKV#3',
'0.604 min SKV#3',
'0.604 max SKV#3',
'0.74 min SKV#3',
'0.74 max SKV#3',
'4,96 SKV#3',
'4,40 SKV#3',
'5,36 SKV#3',
'5,92 SKV#3',
'Sym Einstich SKV#3',
'Sym aussen SKV#3',
'Licht-Einstich frei SKV#3',
'4.713 SKV#4',
'0.604 min SKV#4',
'0.604 max SKV#4',
'0.74 min SKV#4',
'0.74 max SKV#4',
'4,96 SKV#4',
'4,40 SKV#4',
'5,36 SKV#4',
'5,92 SKV#4',
'Sym Einstich SKV#4',
'Sym aussen SKV#4',
'Licht-Einstich frei SKV#4',
# 'timestamp',
'Beacon INT0028 - BME280_RH',
# 'timestamp.1',
'Beacon INT0028 - BMP280_PRESS',
# 'timestamp.2',
'Beacon INT0028 - BMP280_TEMP',
# 'pieces',
# 'tool io',
]

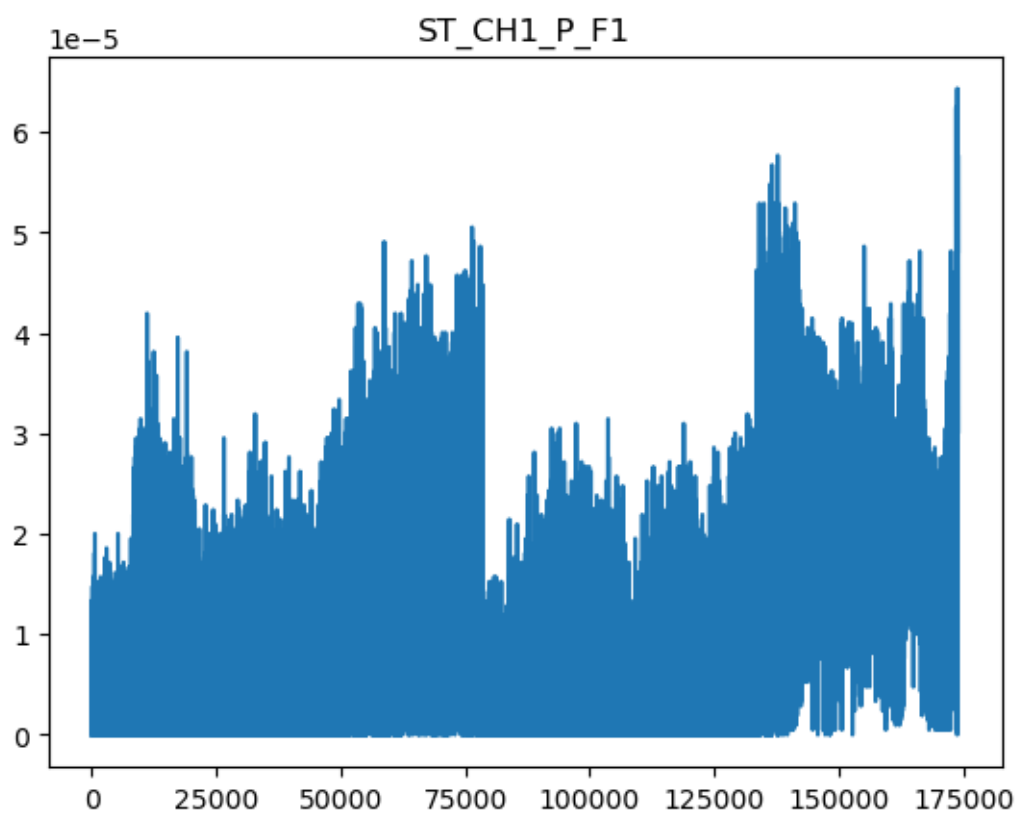
```

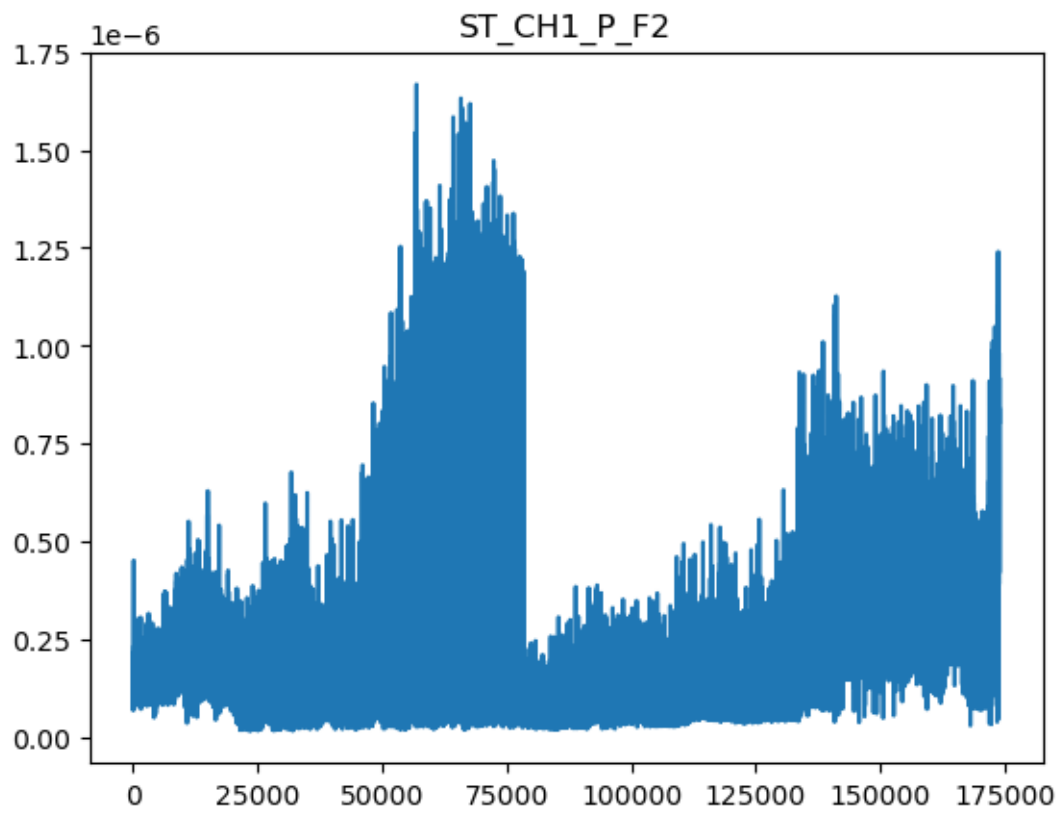
```

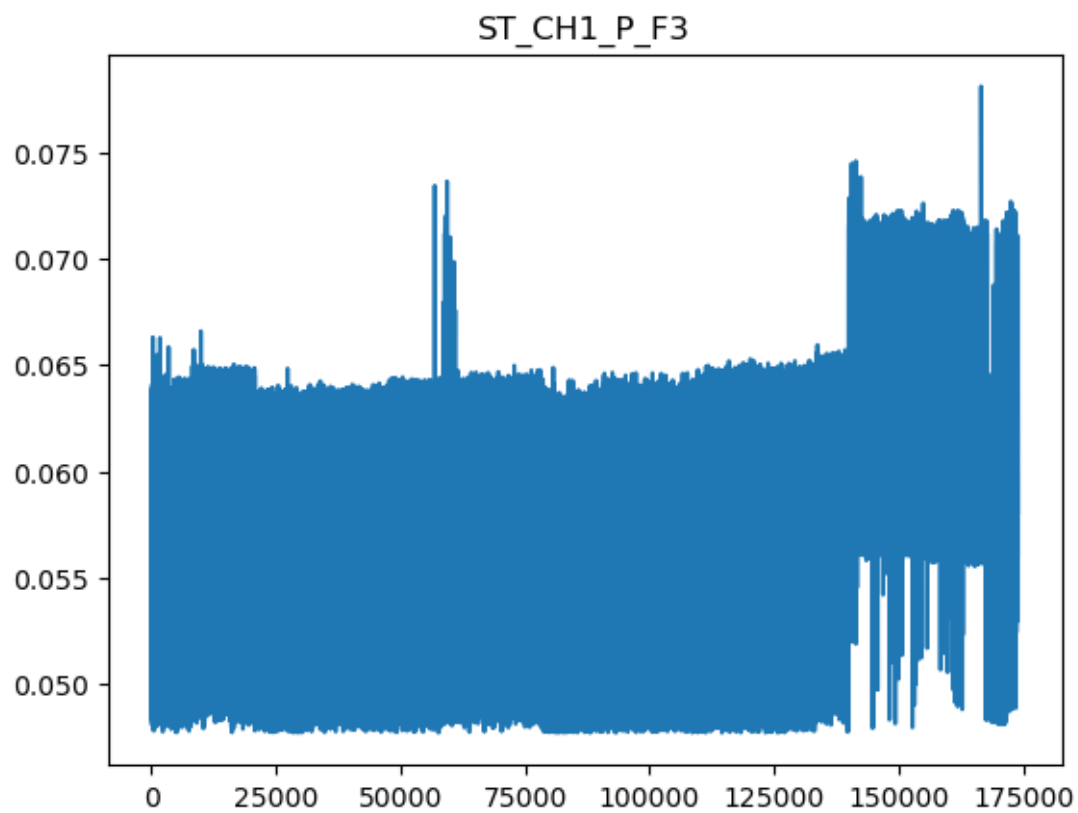
[8]: for f in interestingColumns:
      plt.plot(X.index,X[f])
      plt.title(f)
      plt.show()

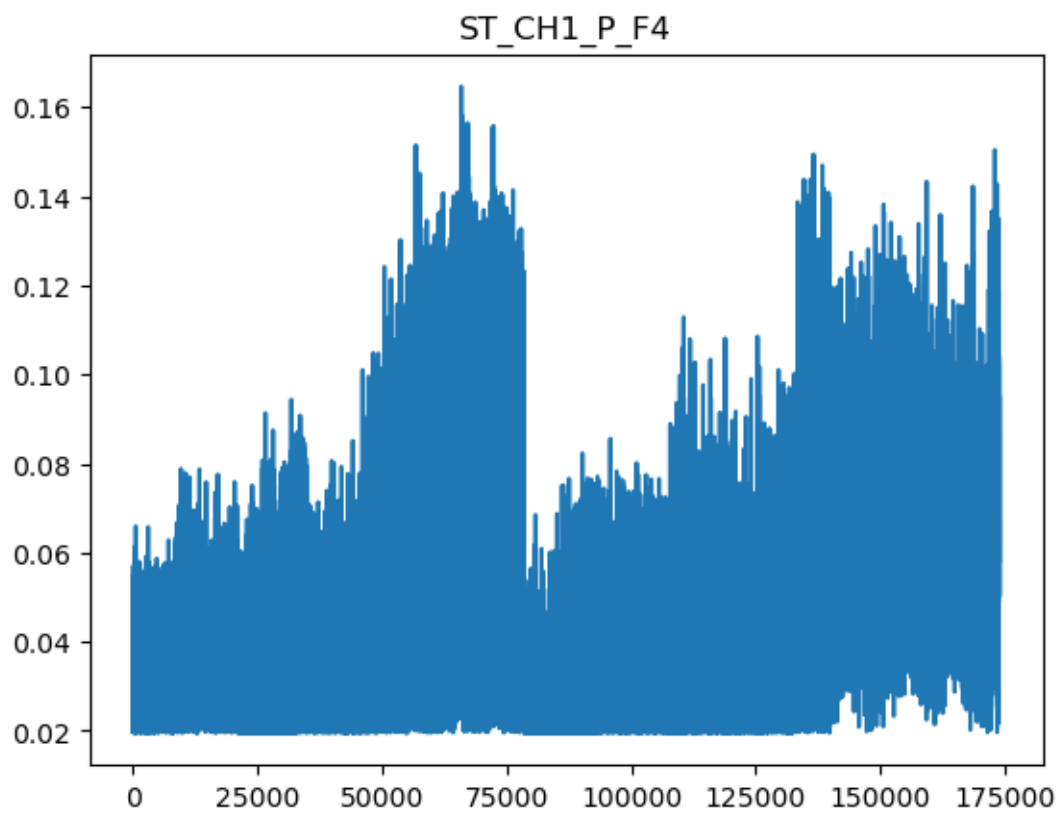
```

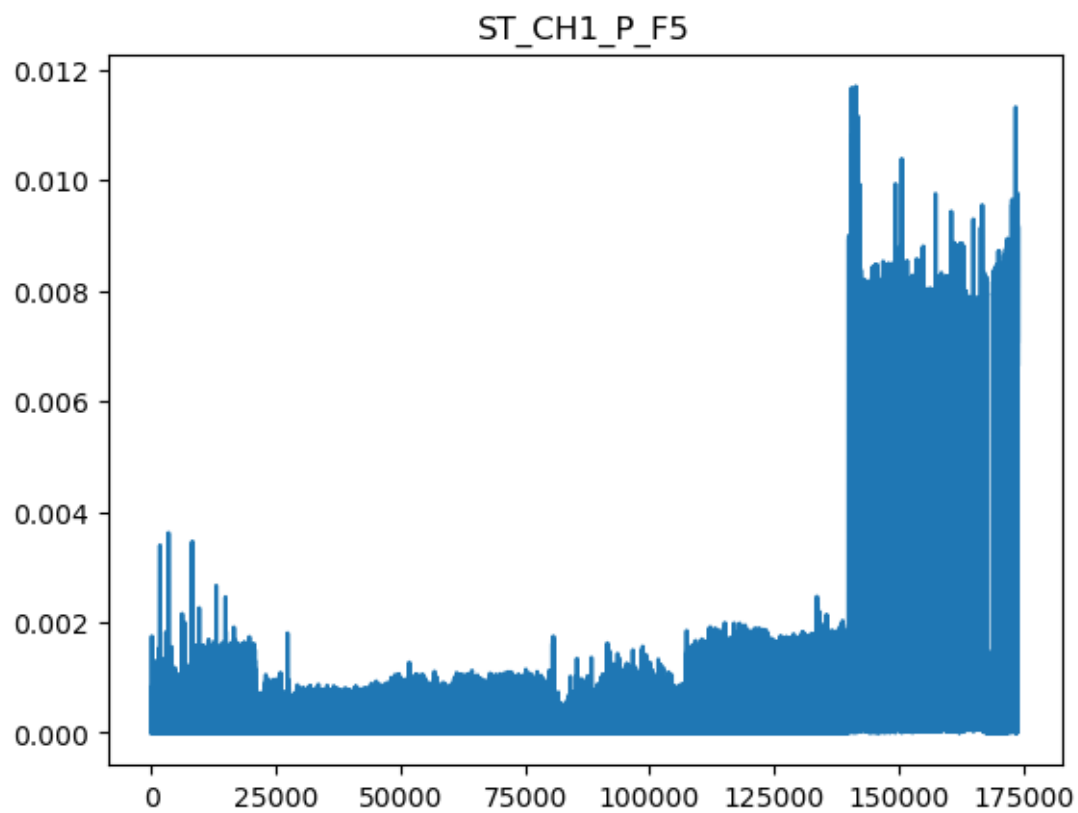


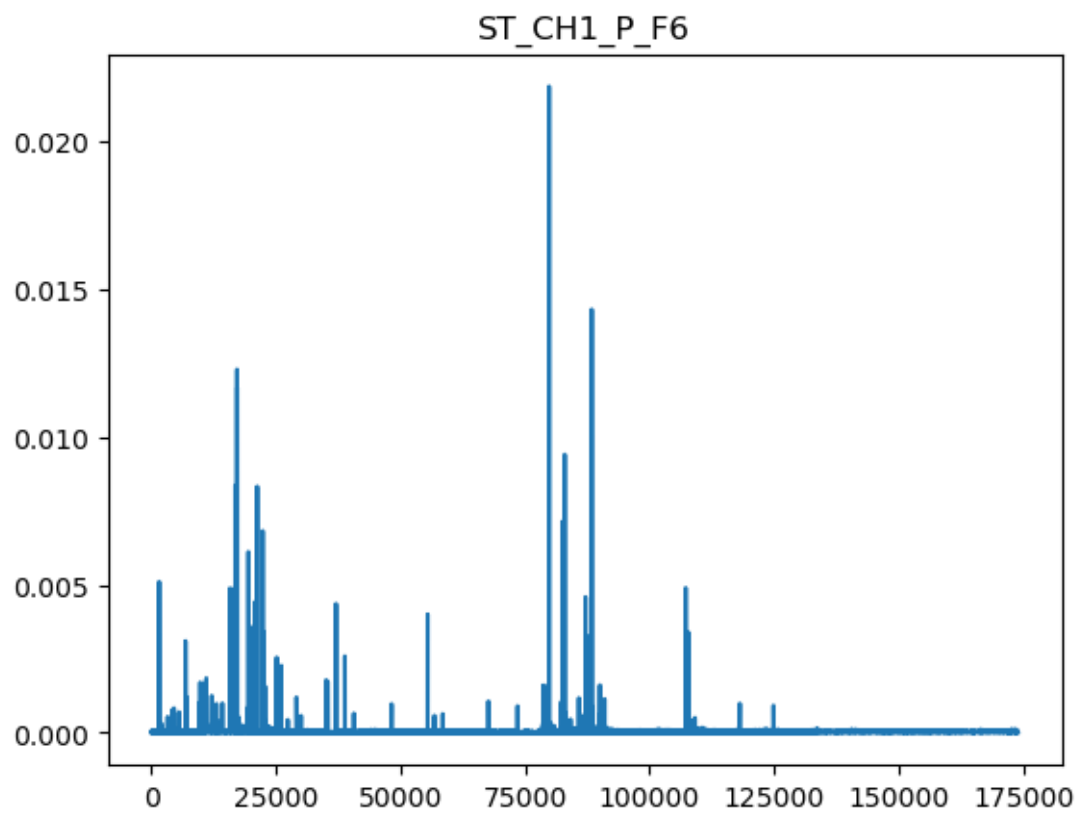




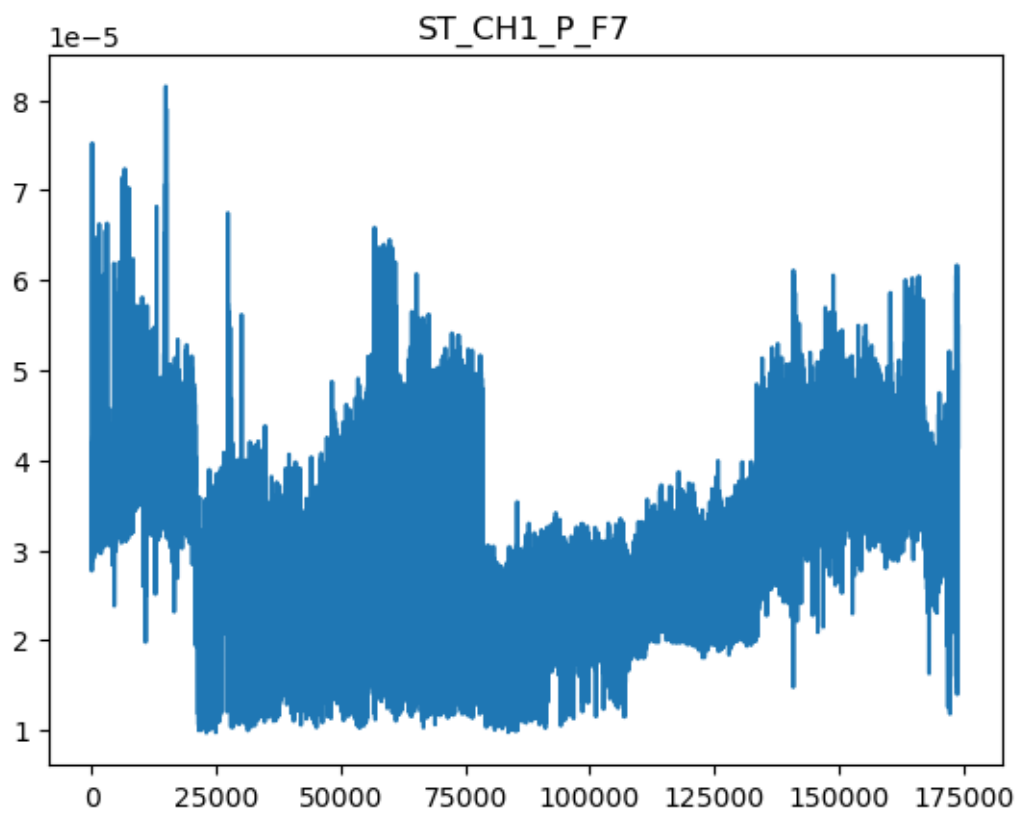


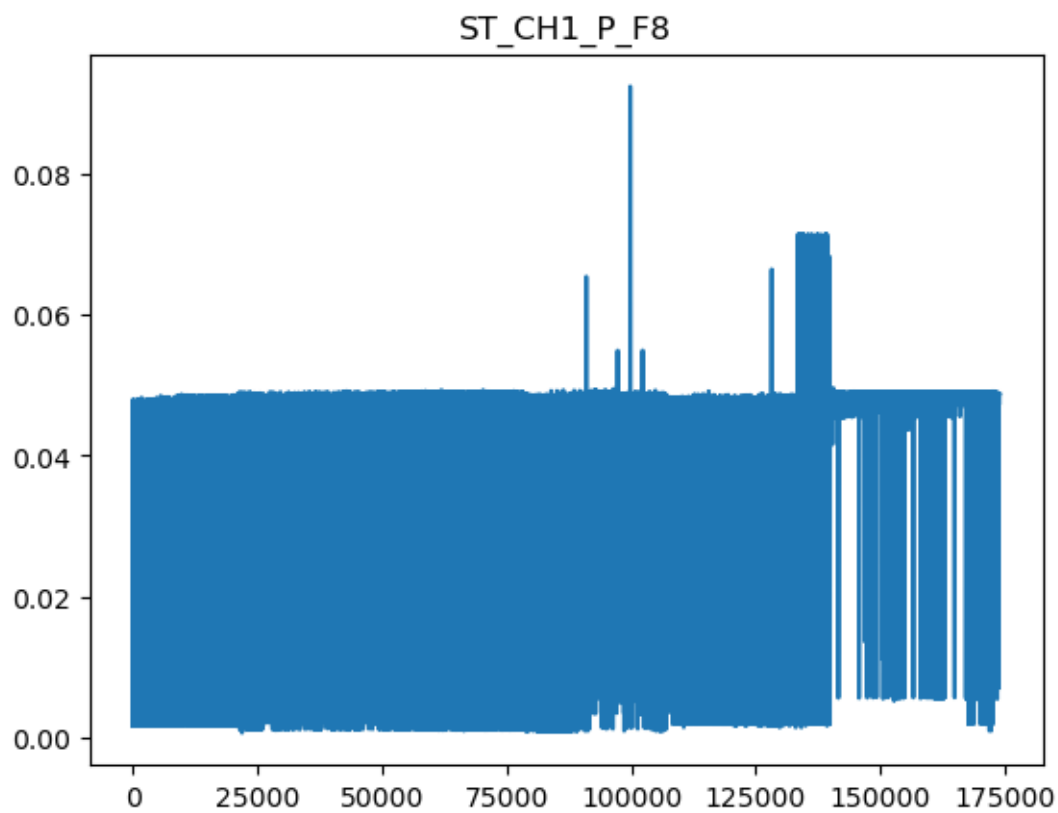


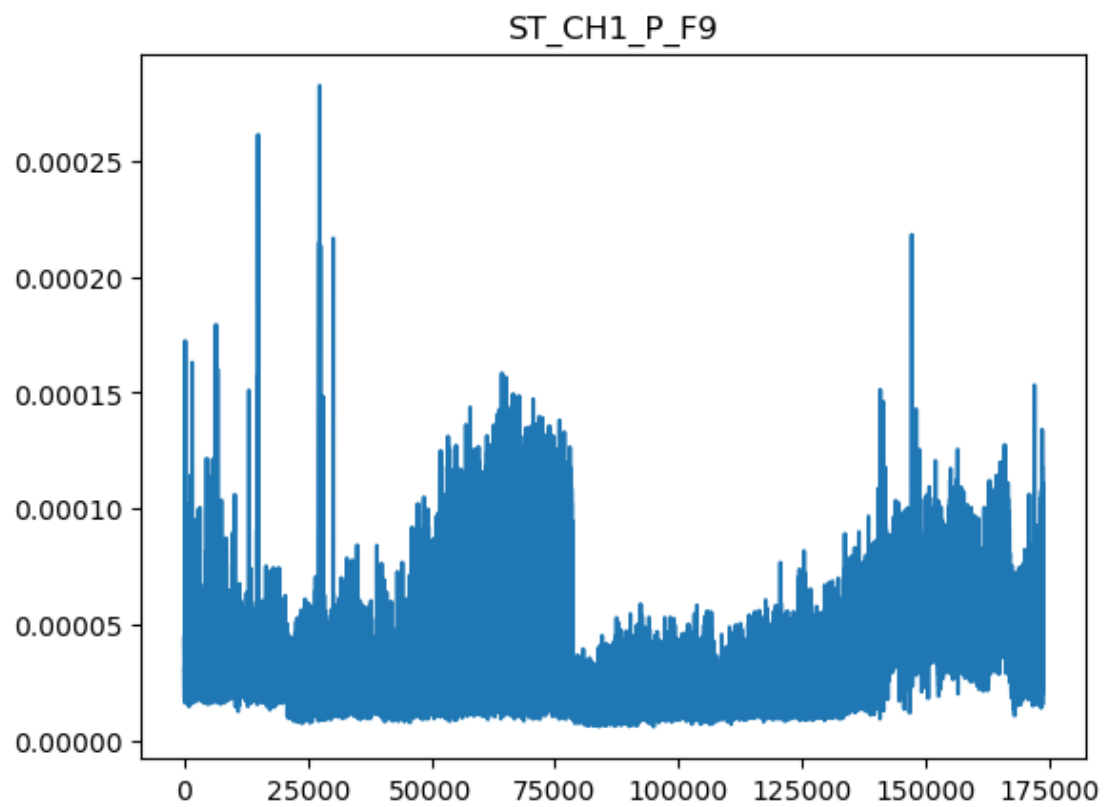


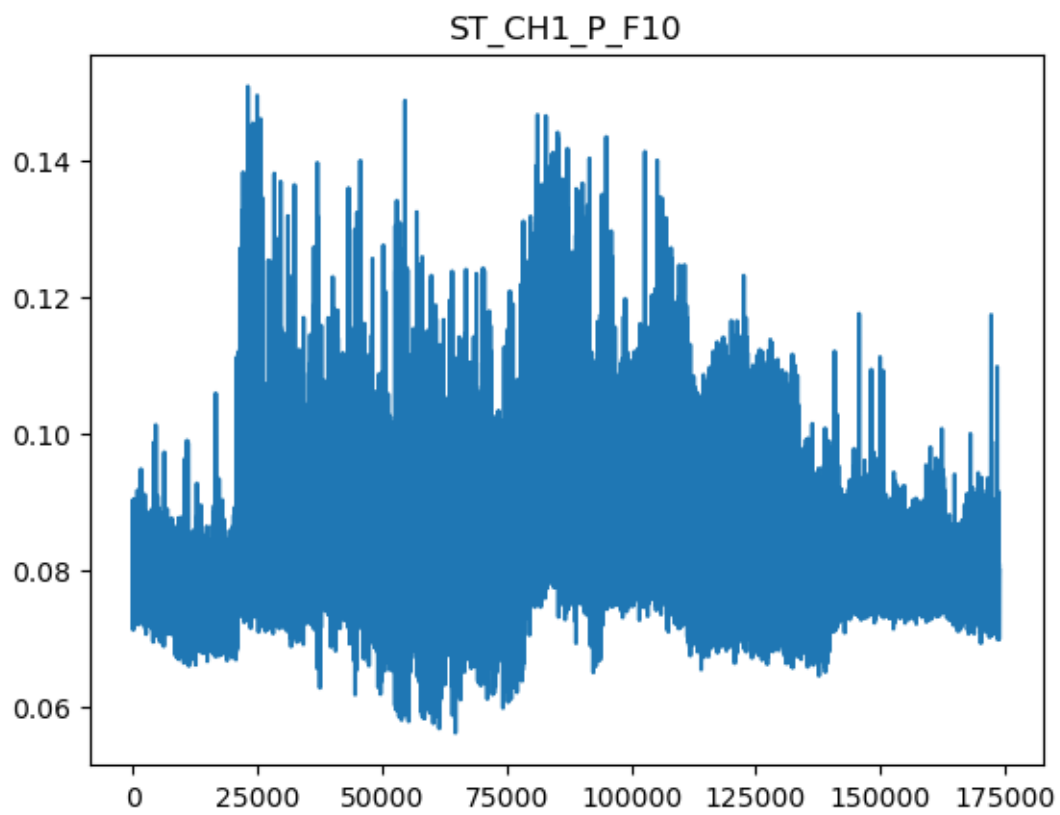


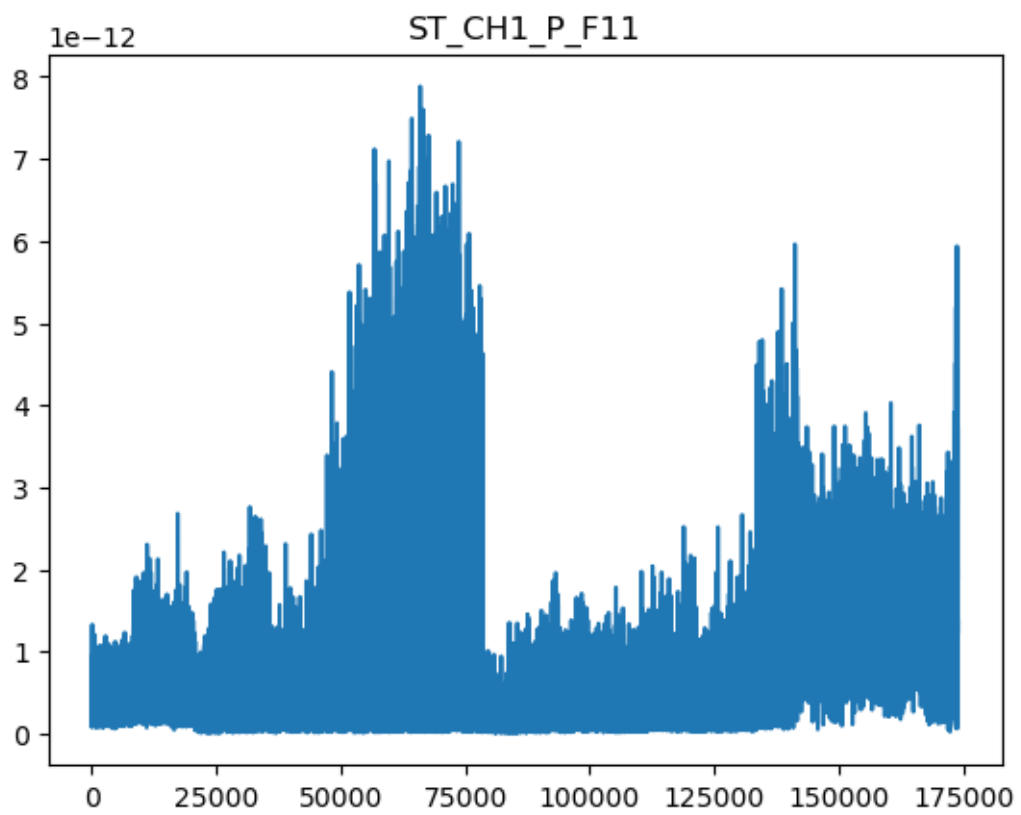


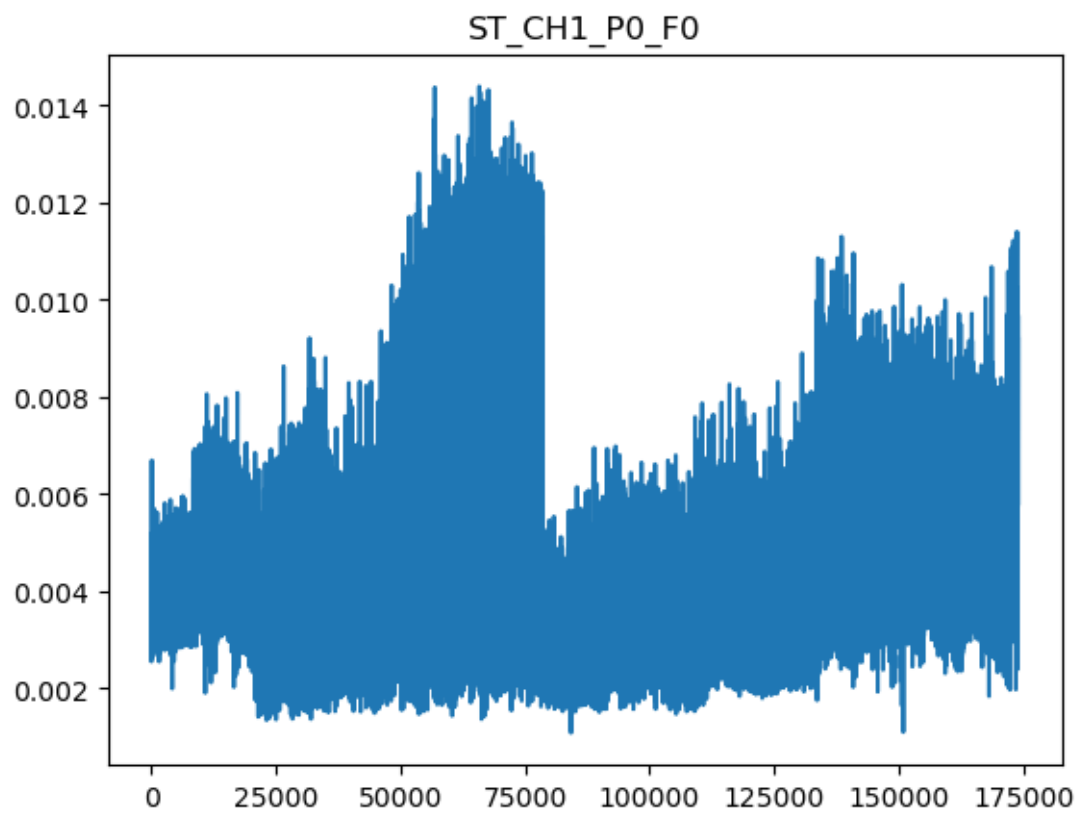


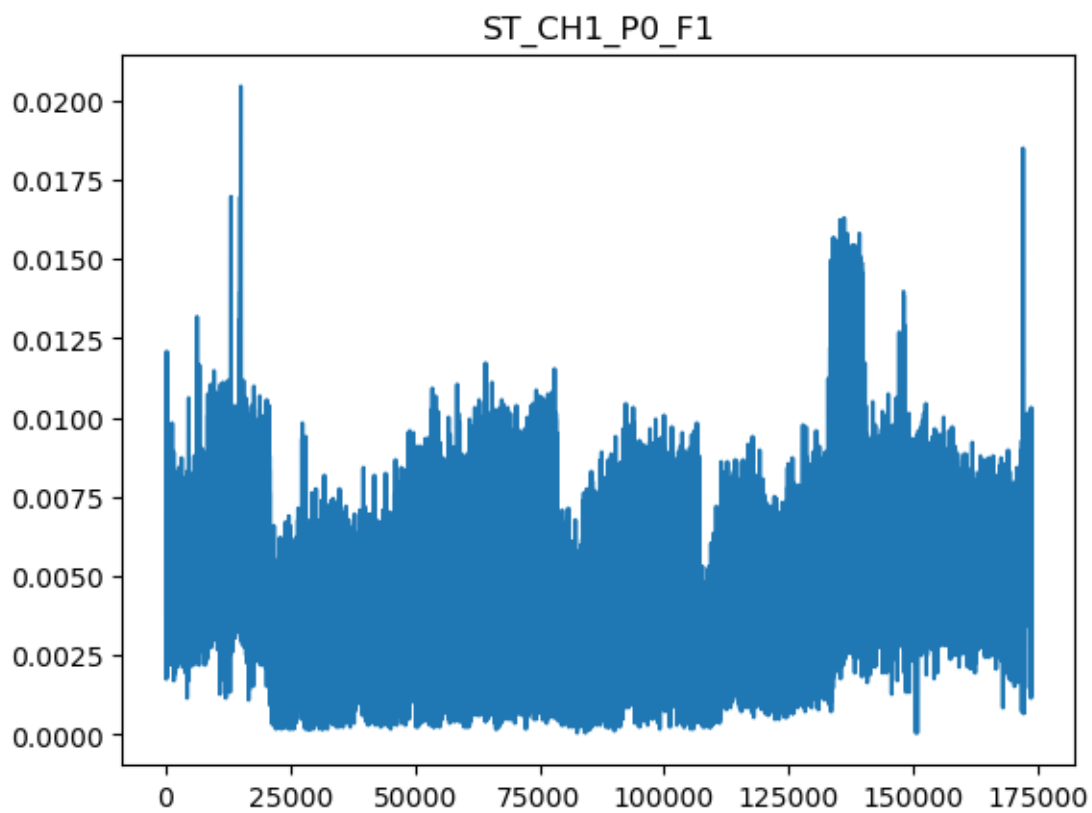


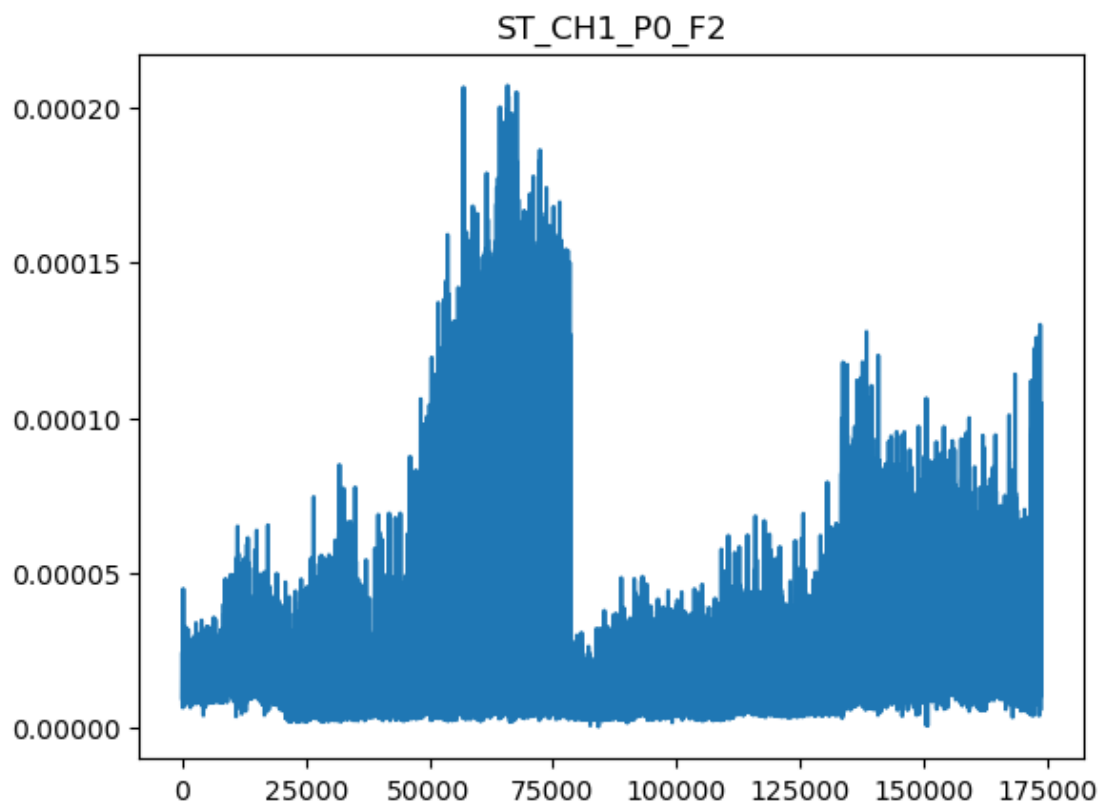




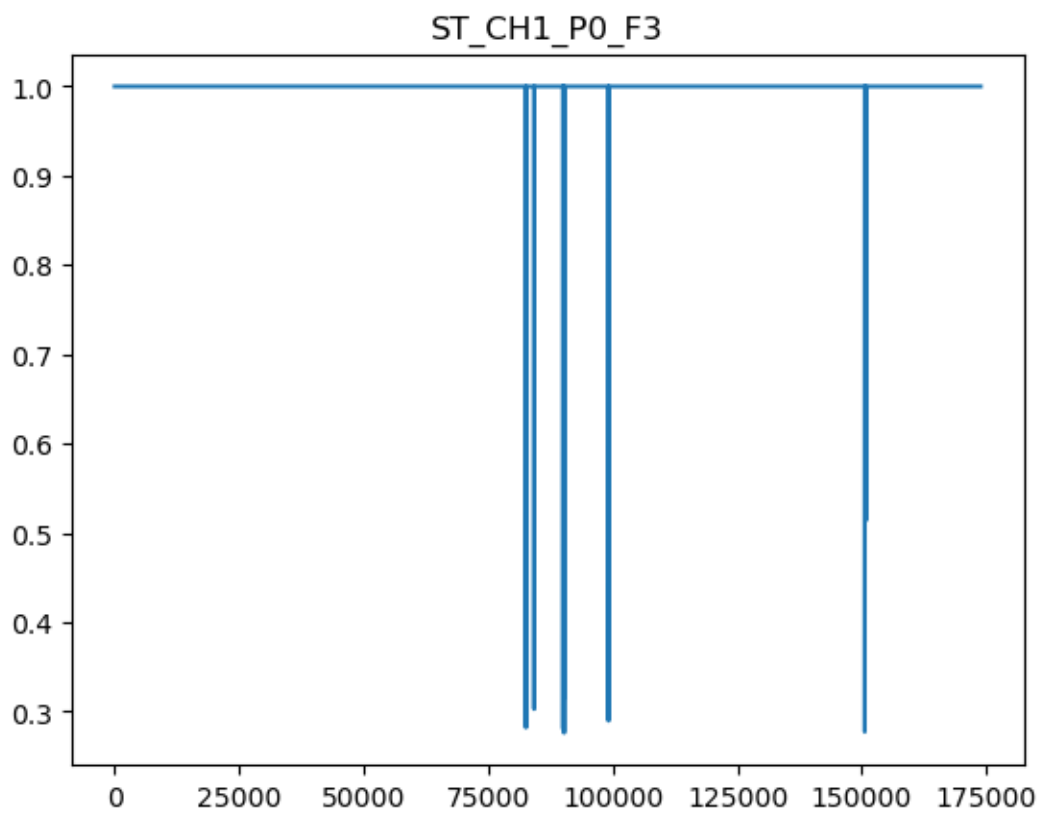


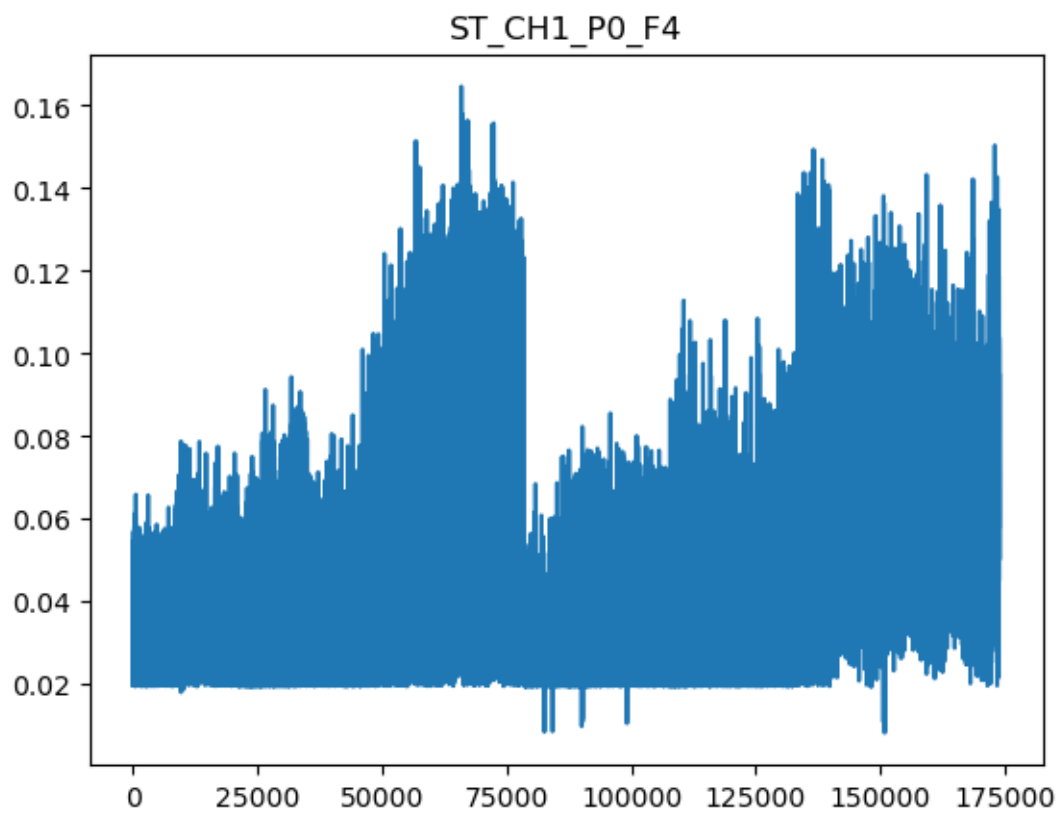


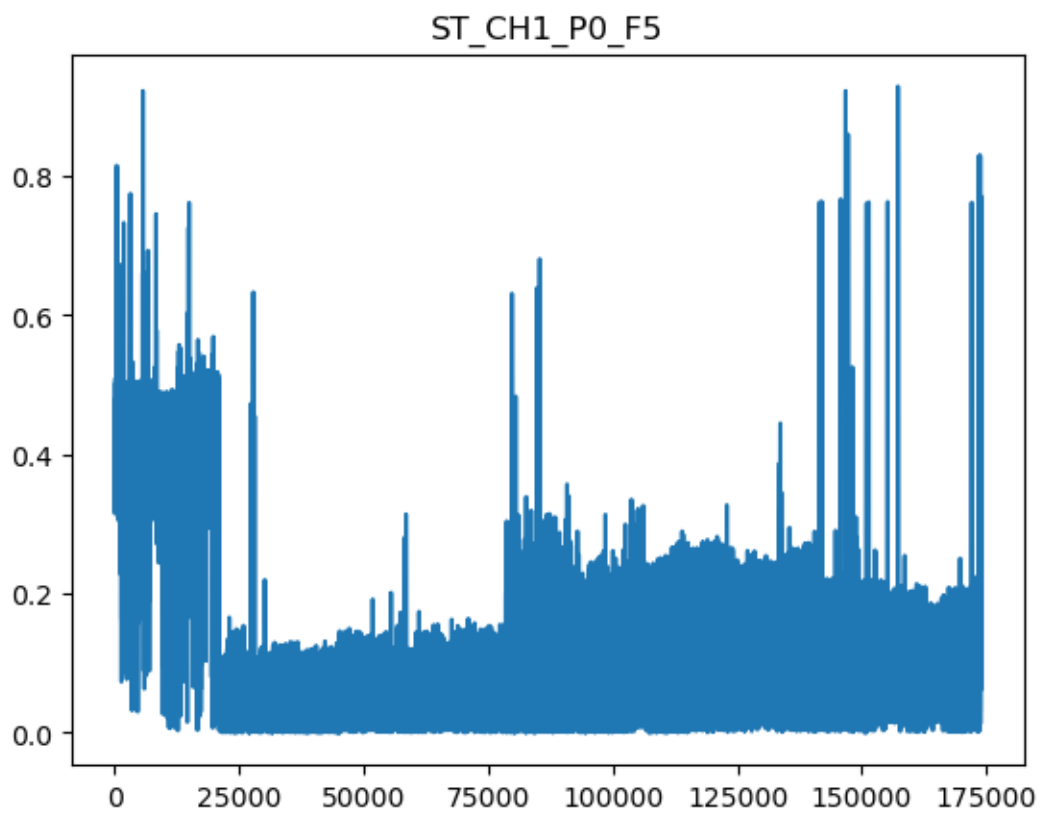


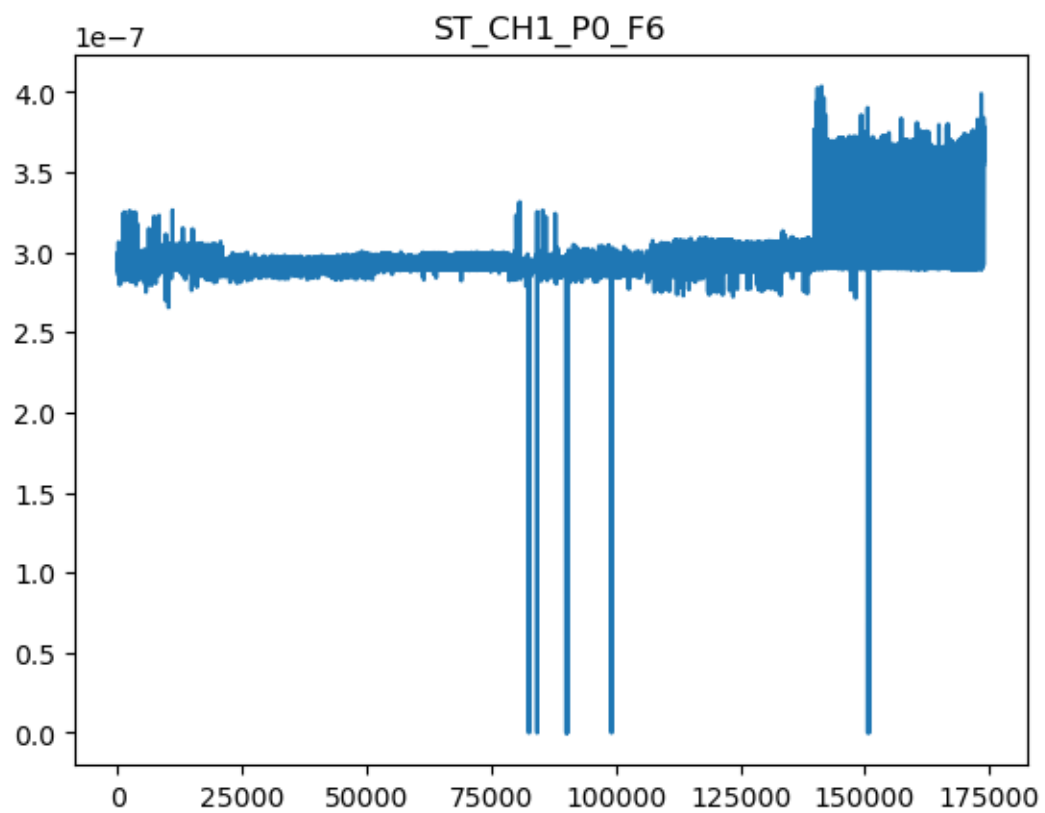


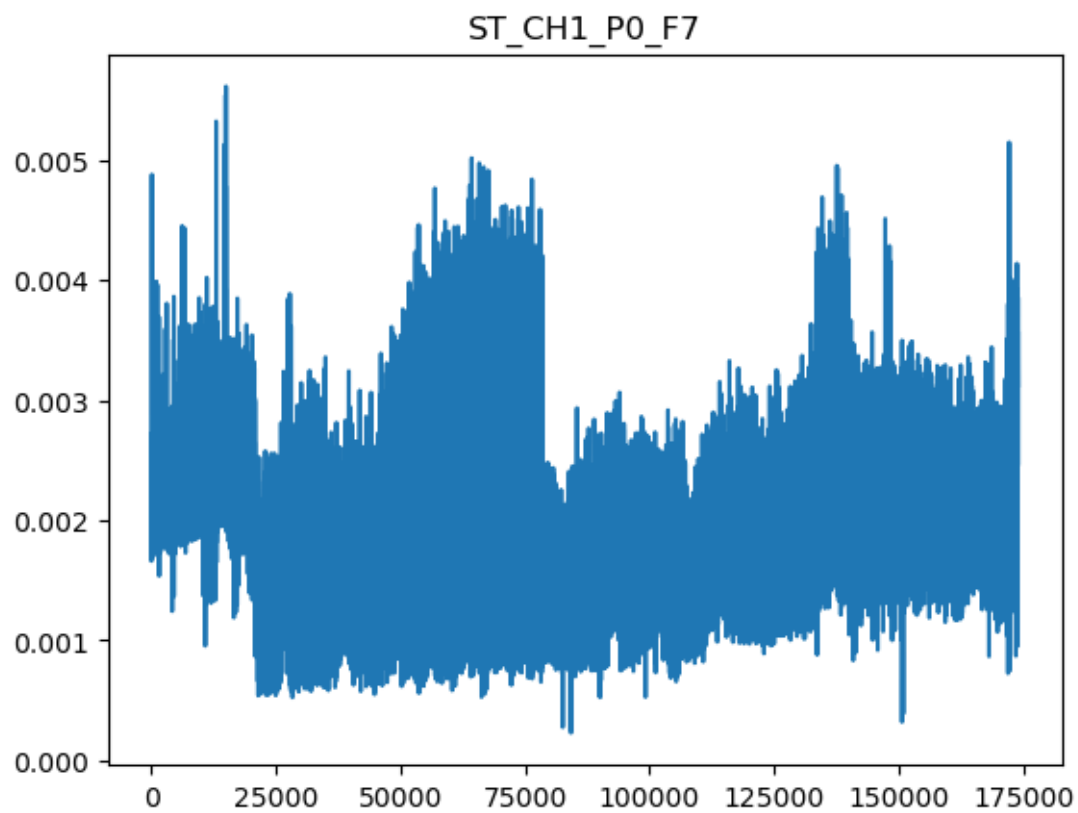


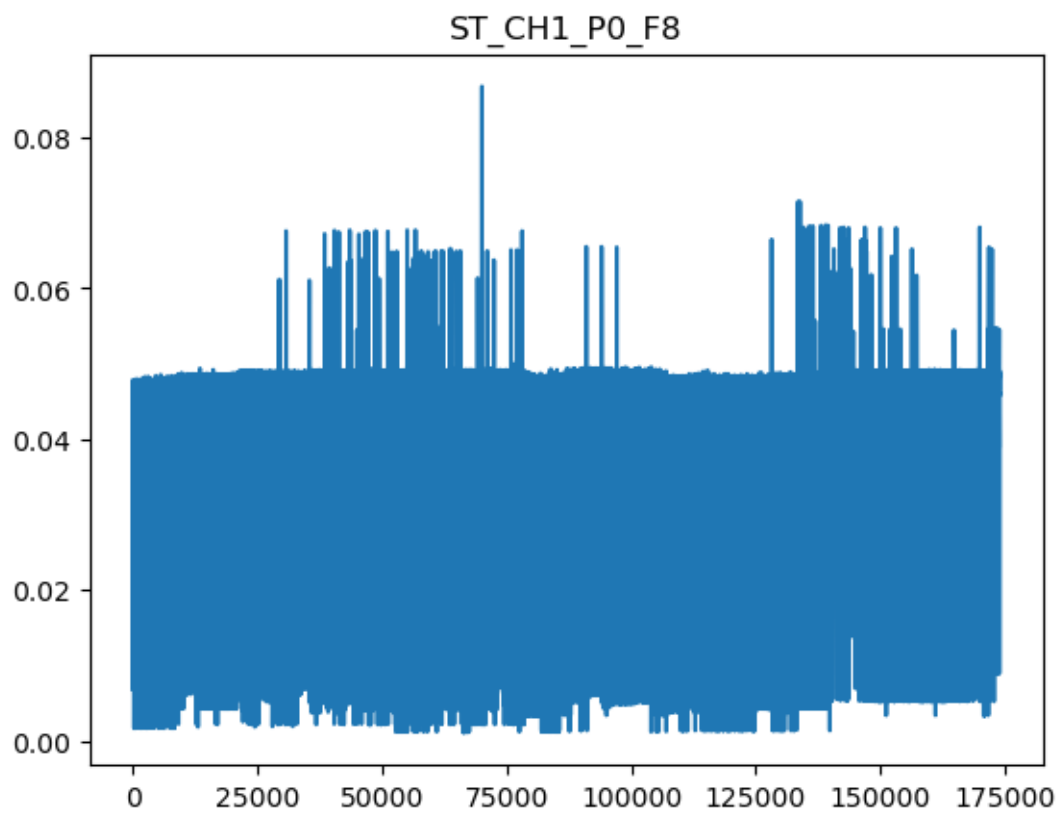


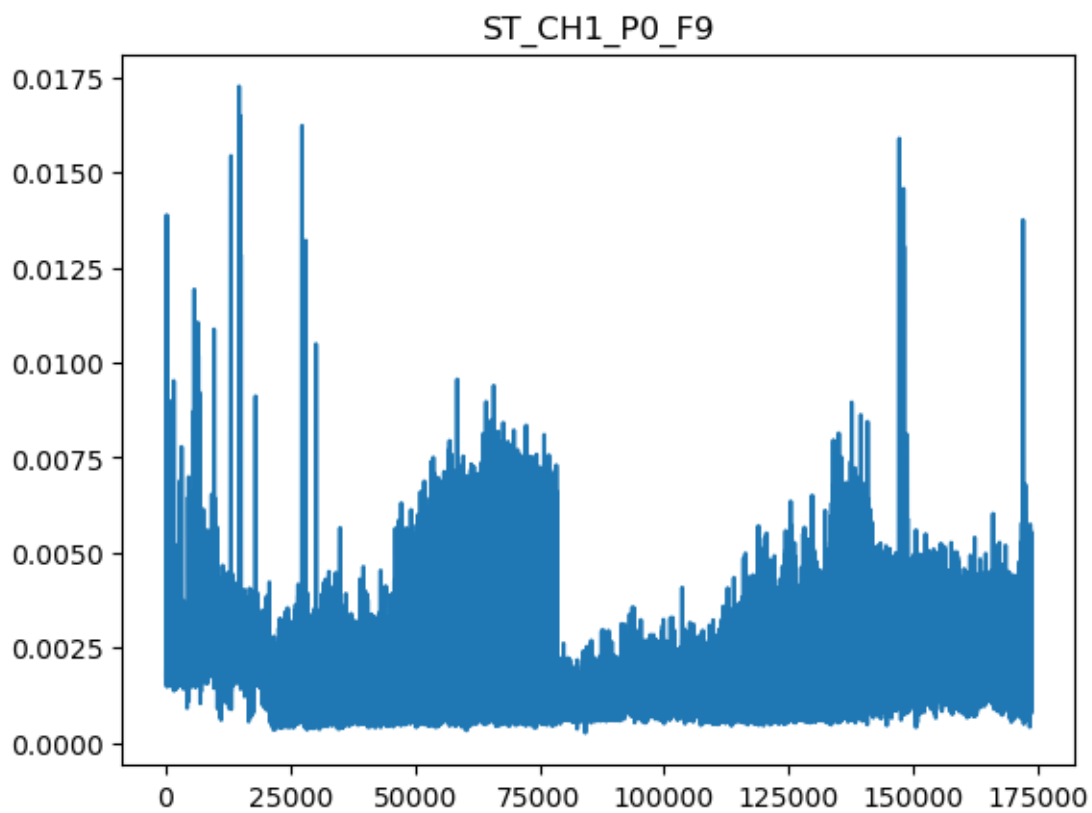


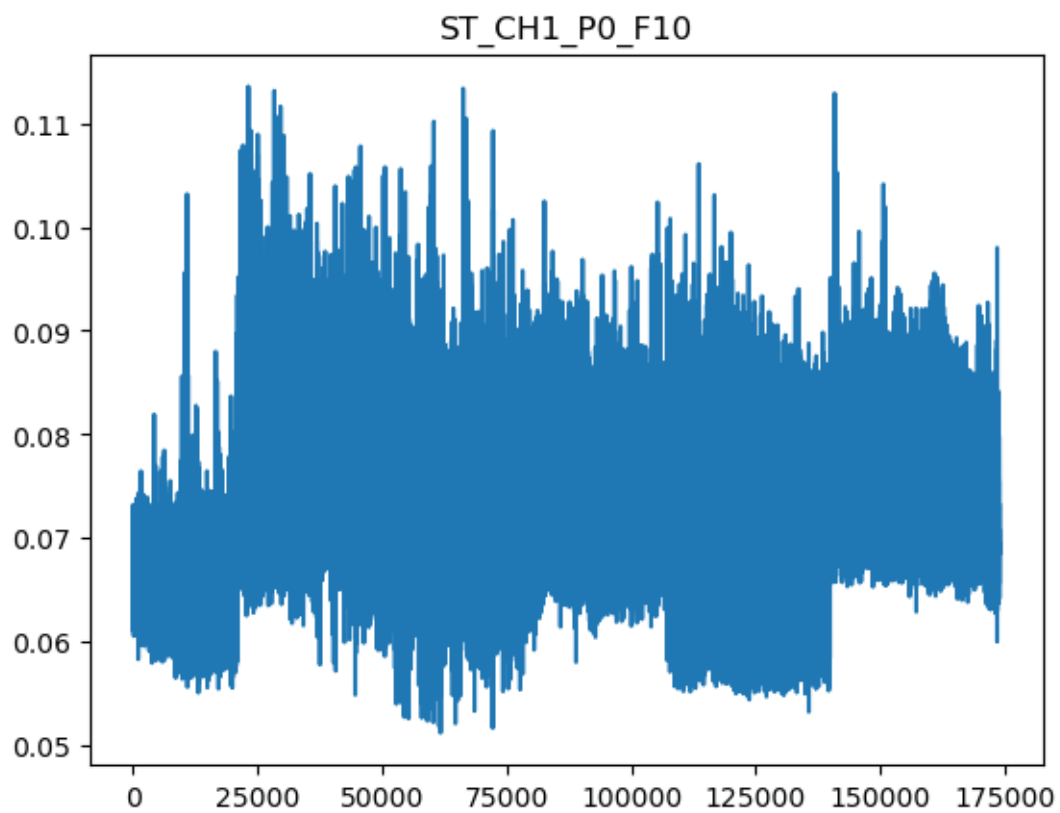




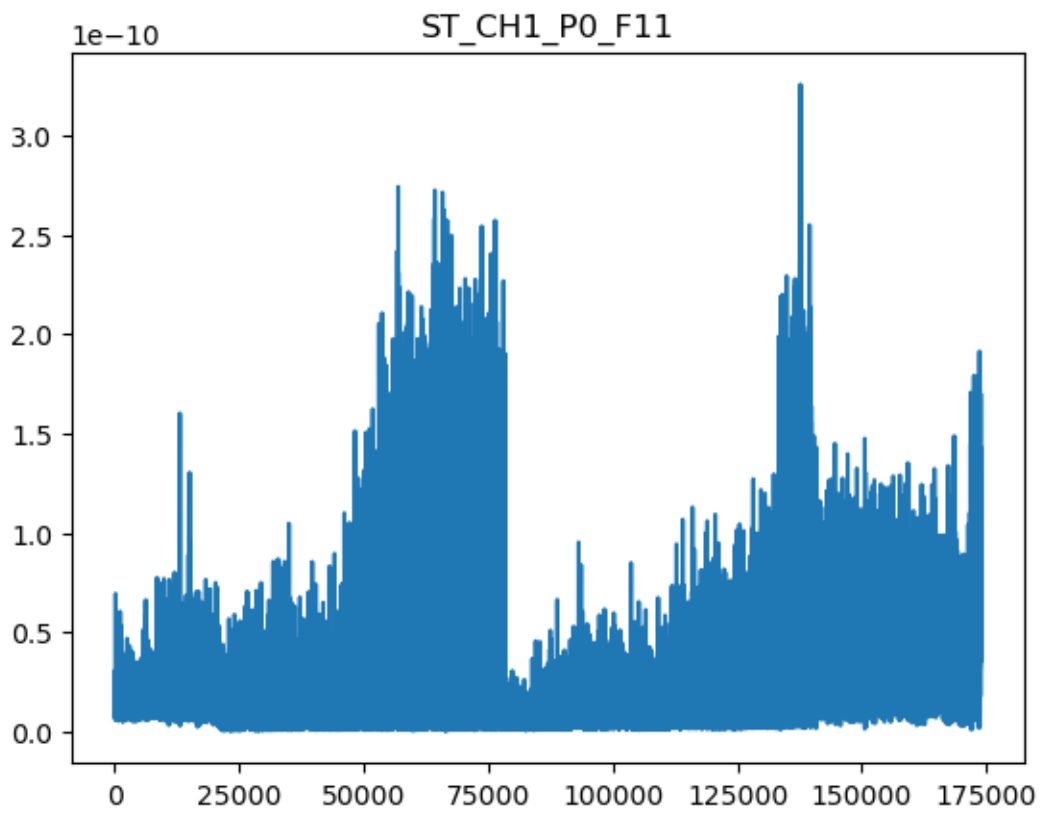


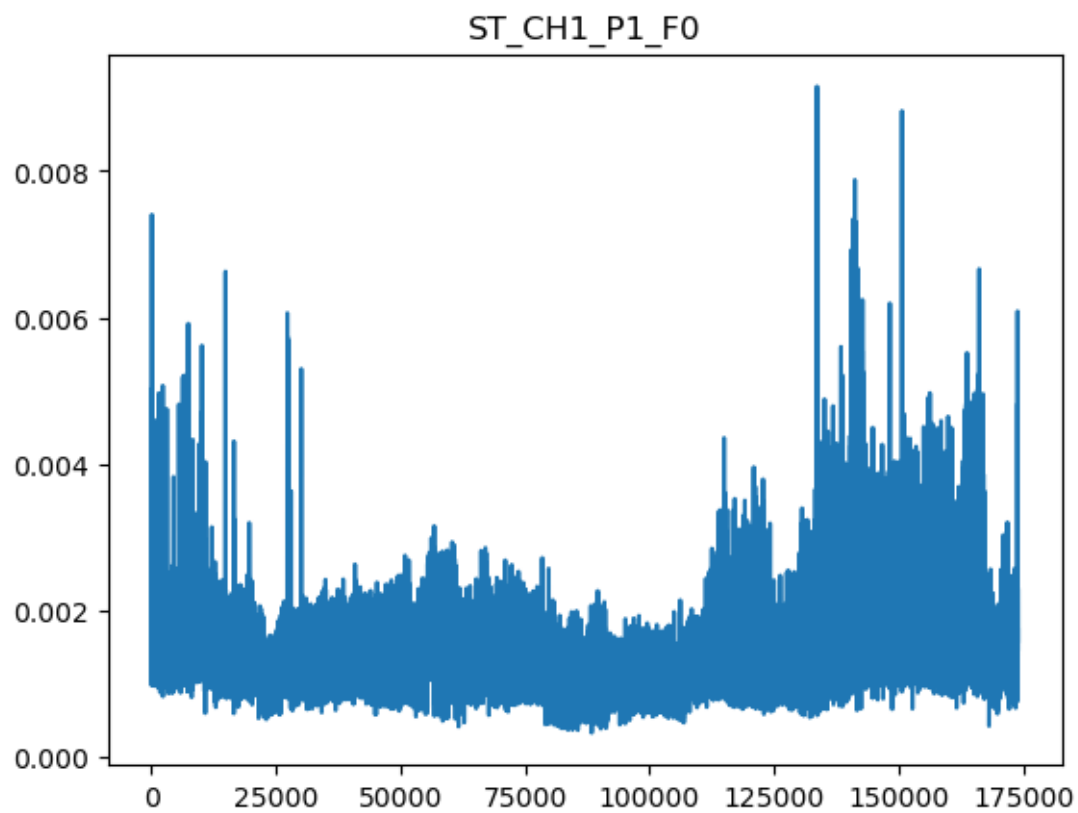


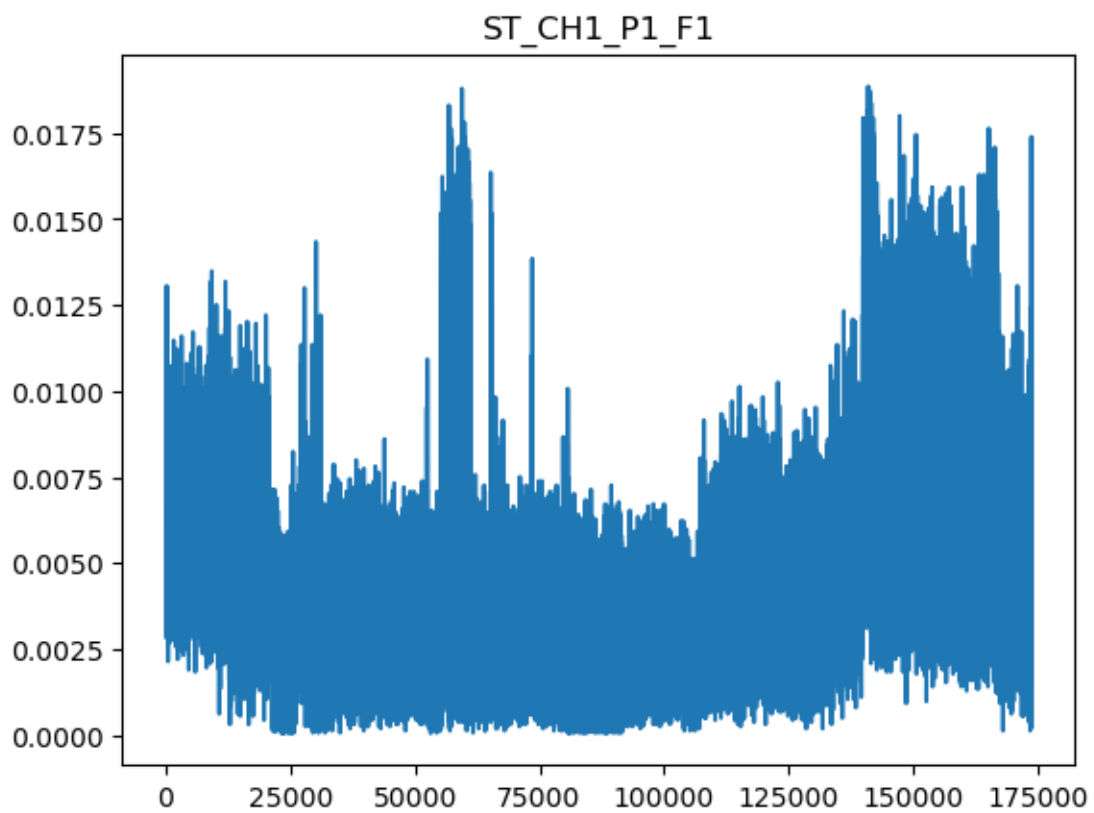


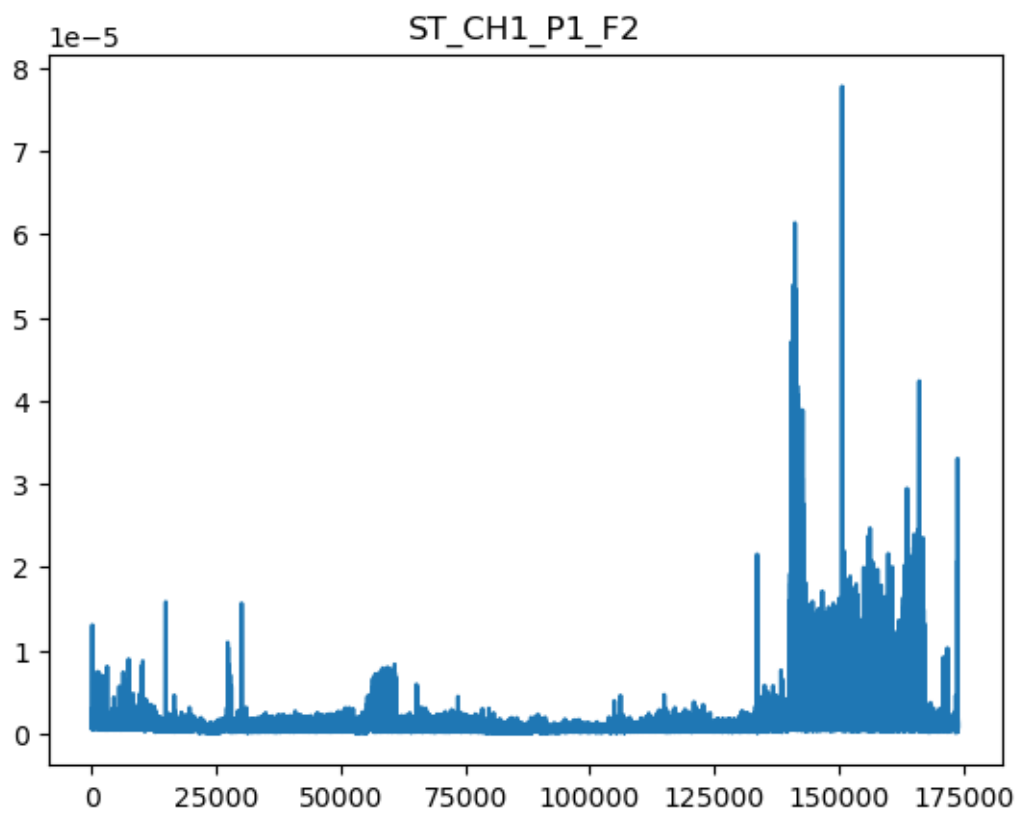


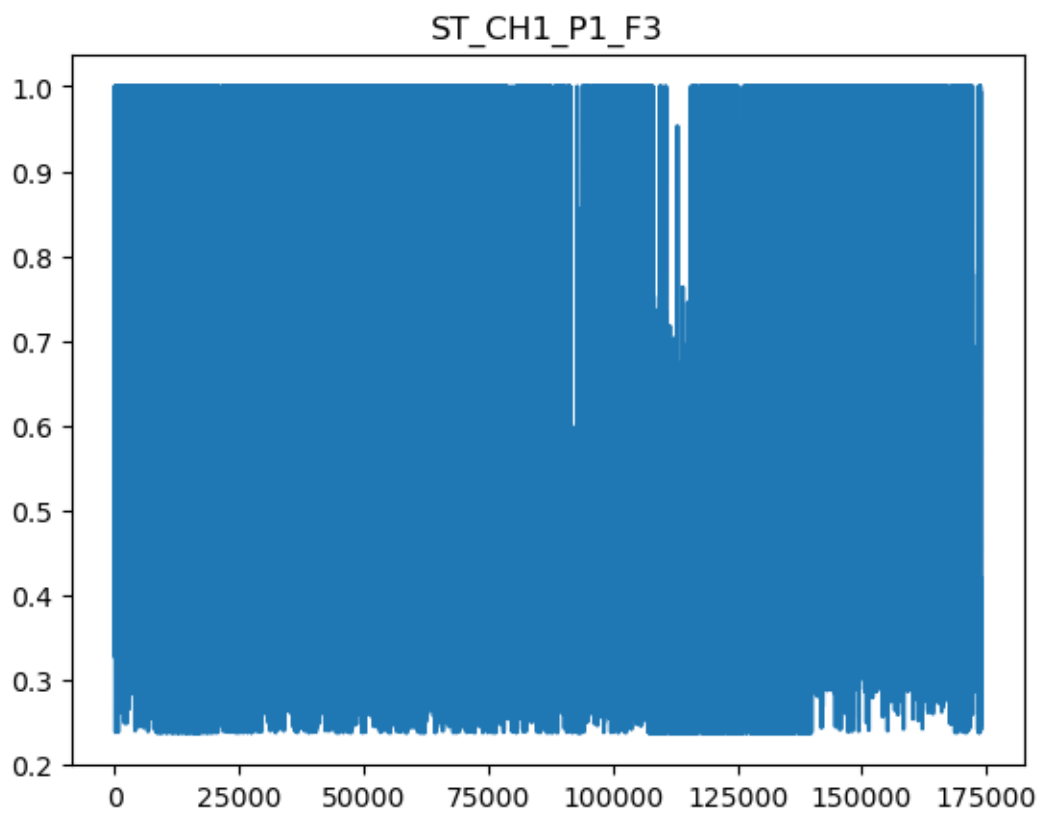


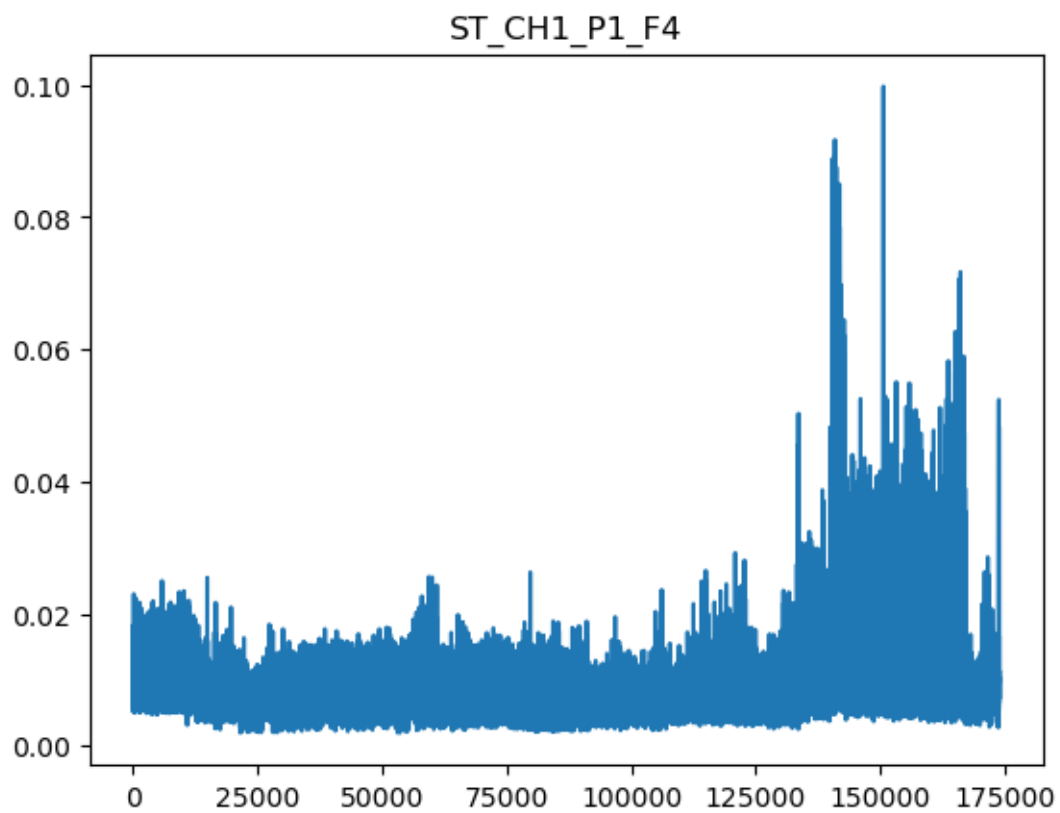


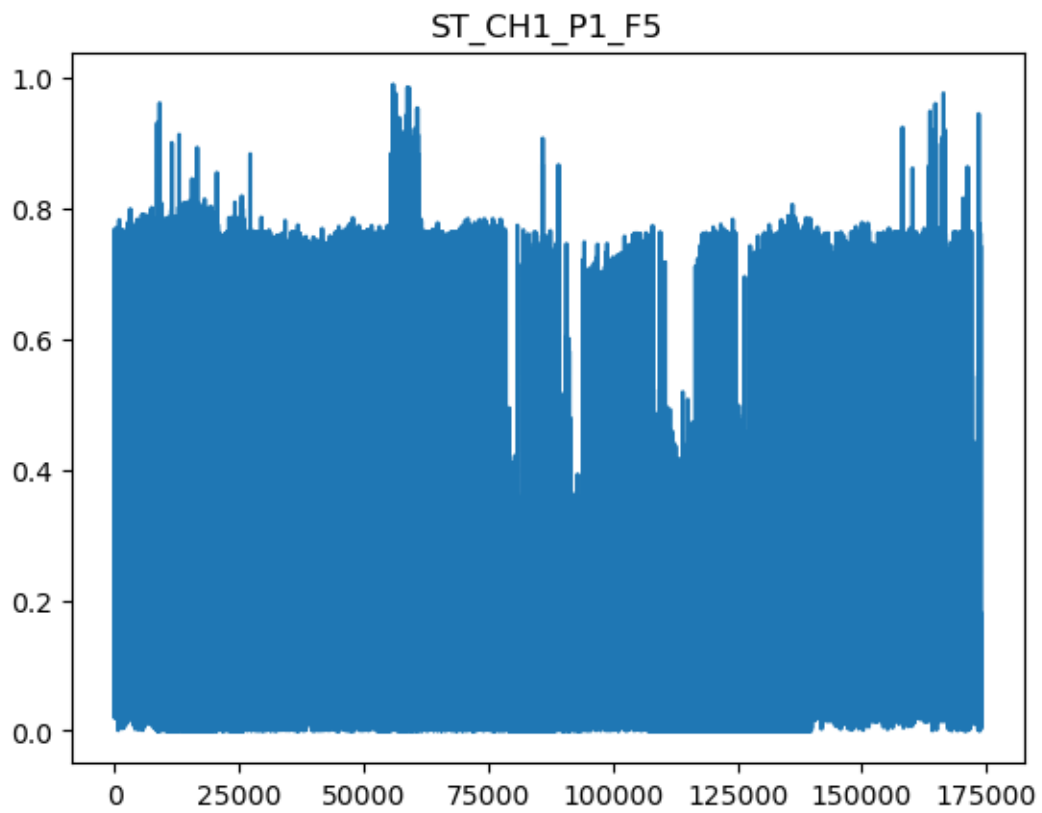


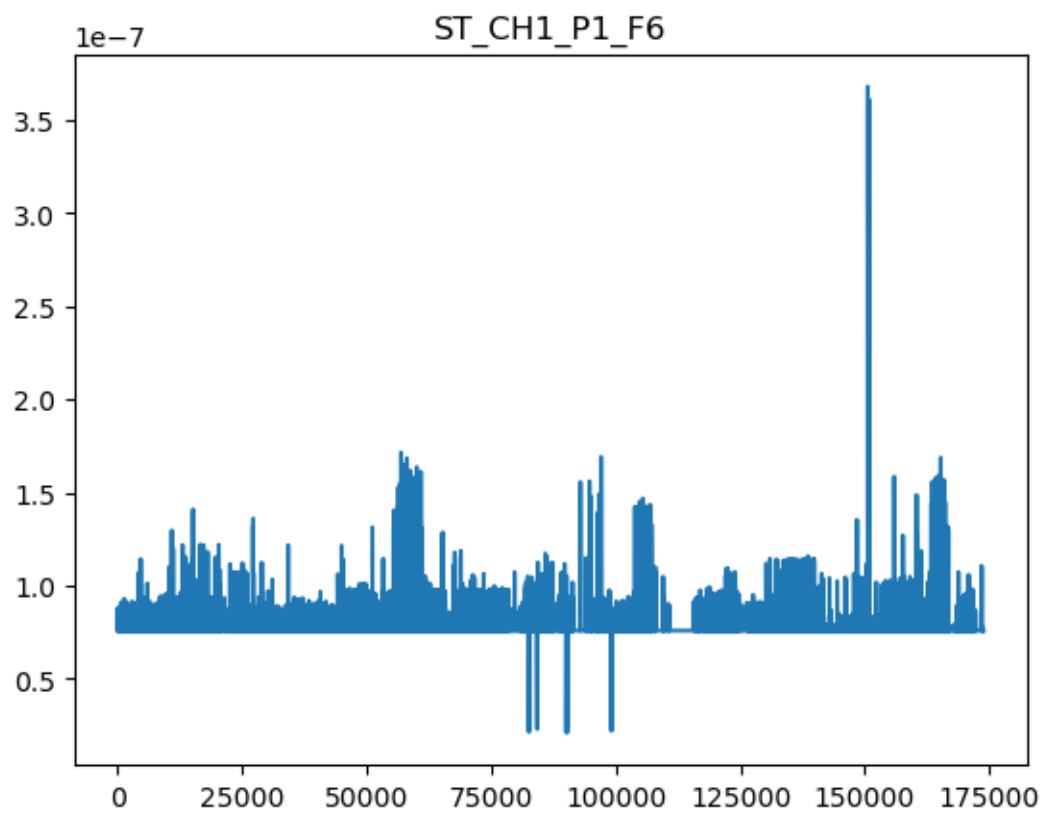




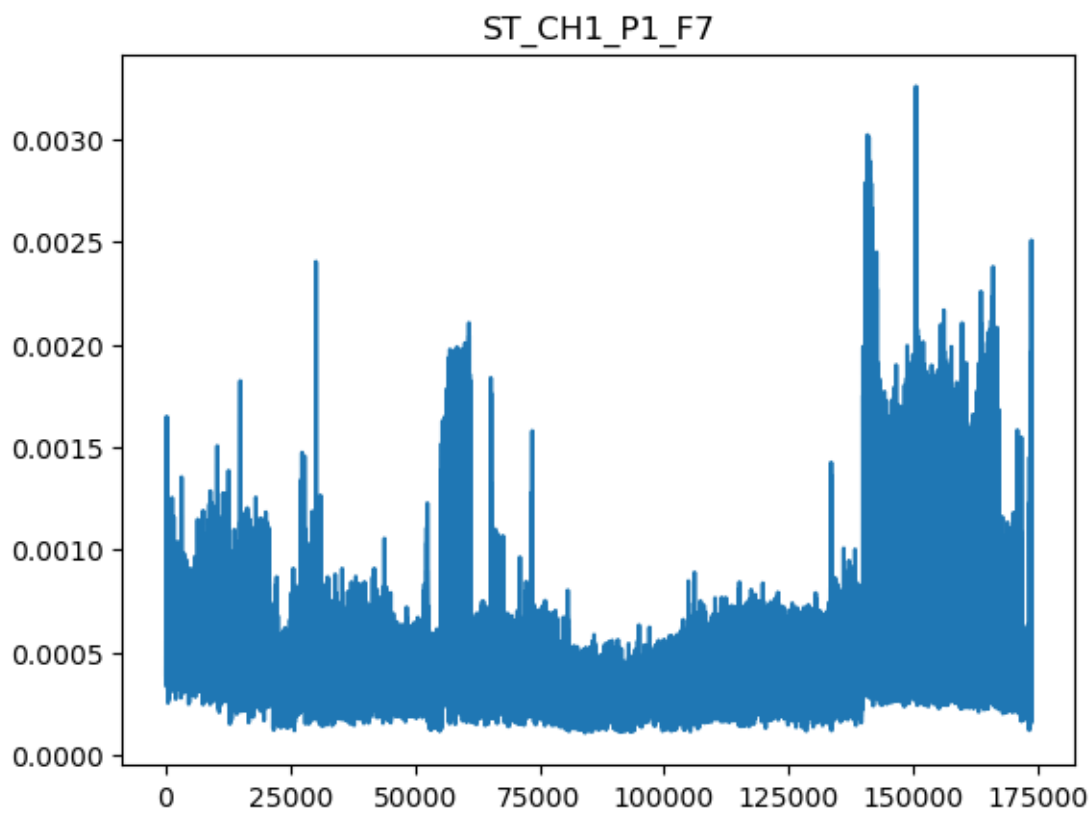


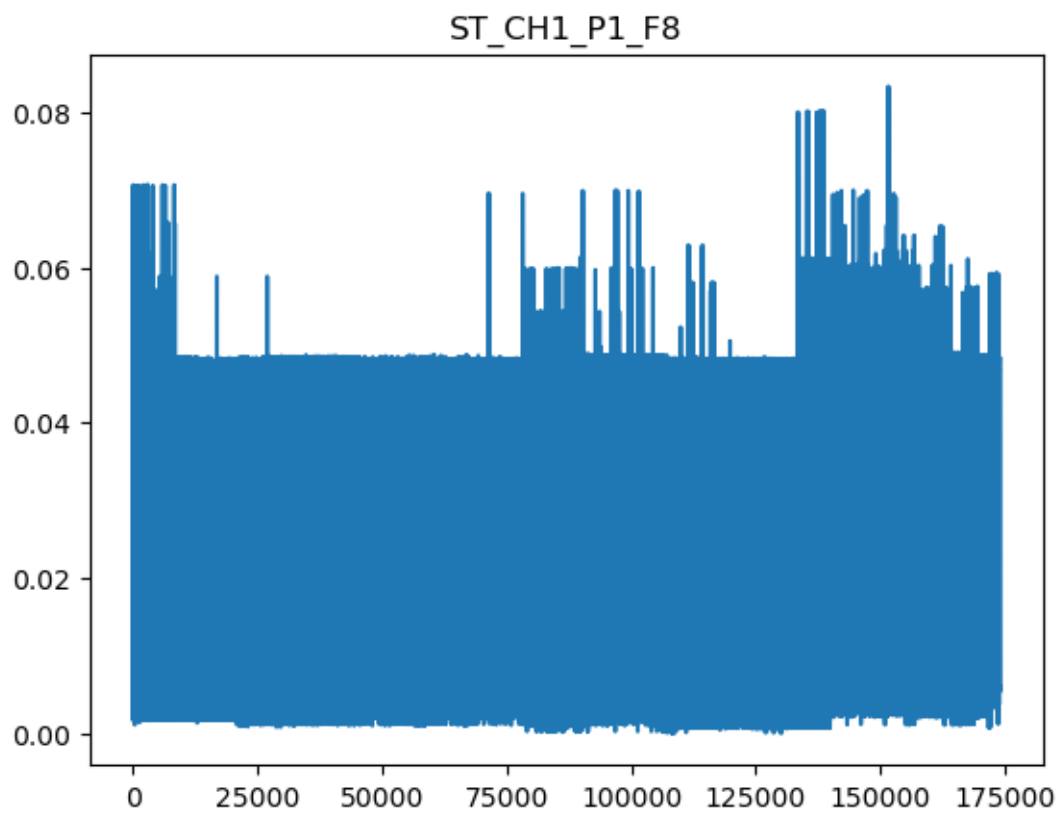


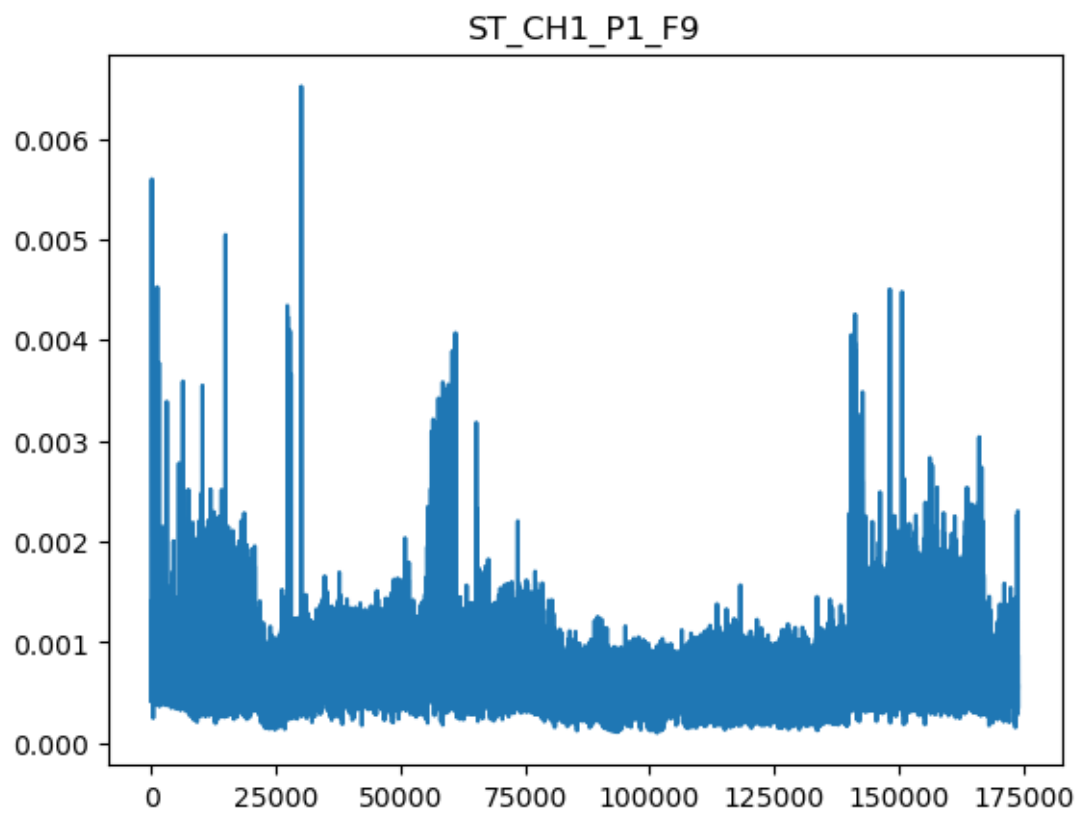


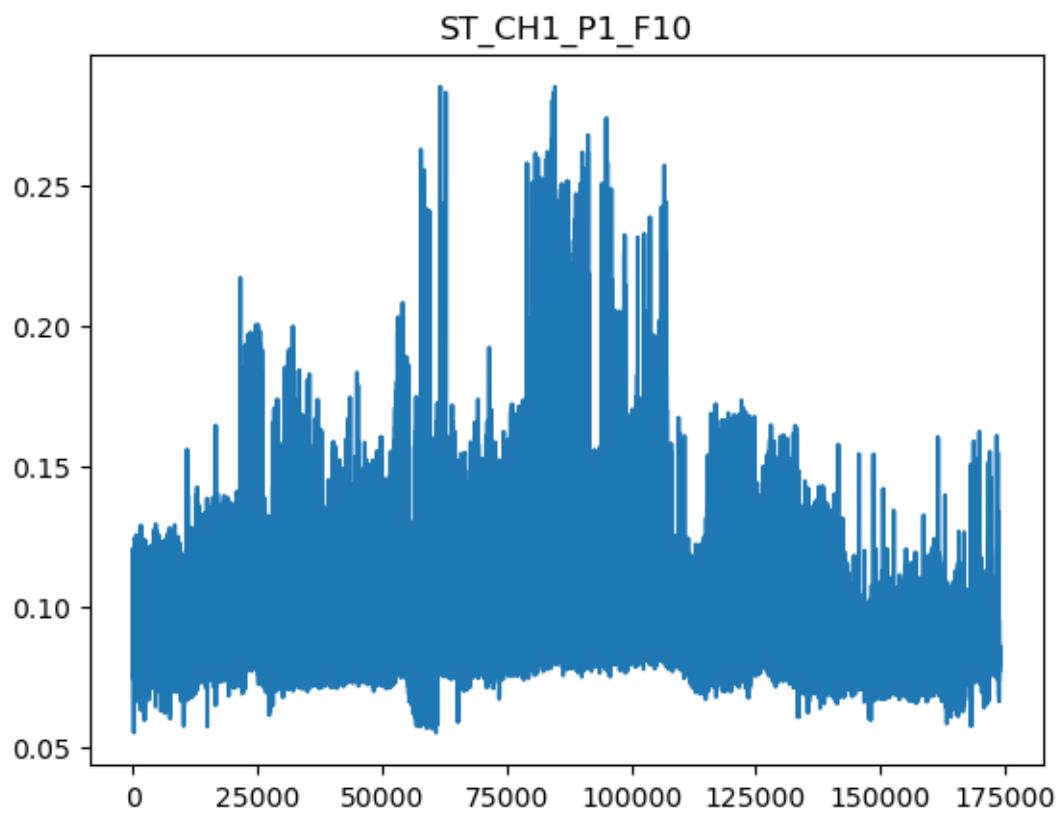


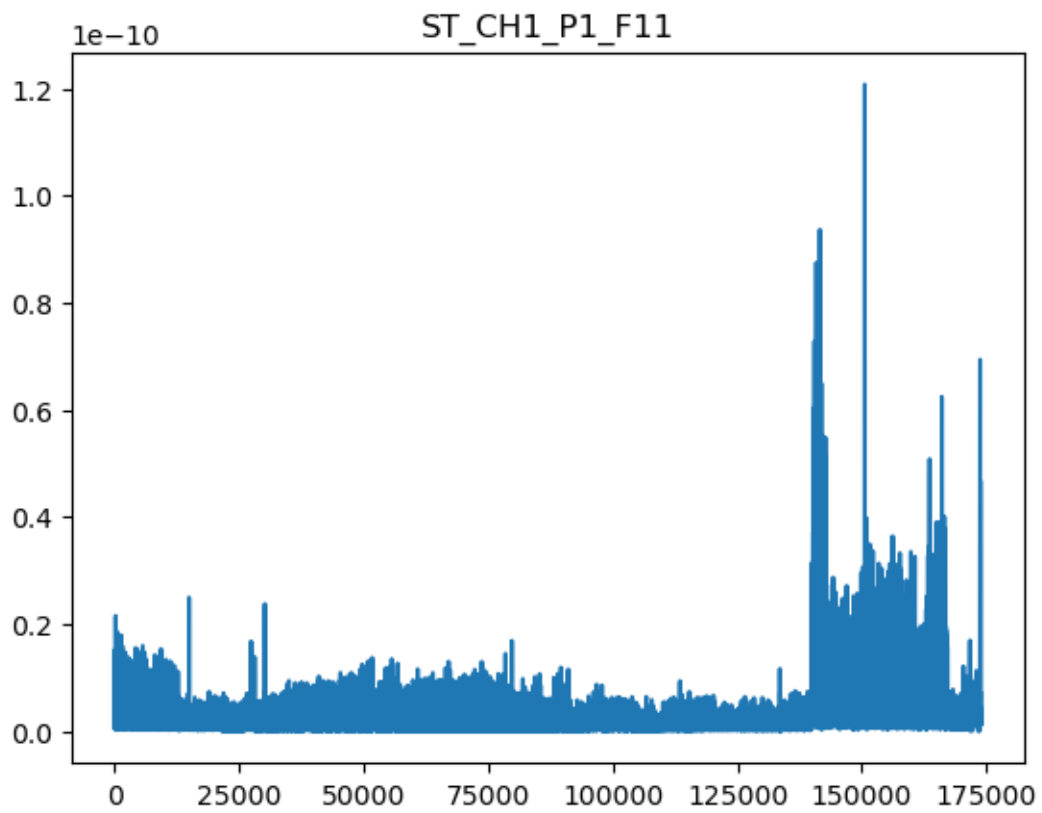


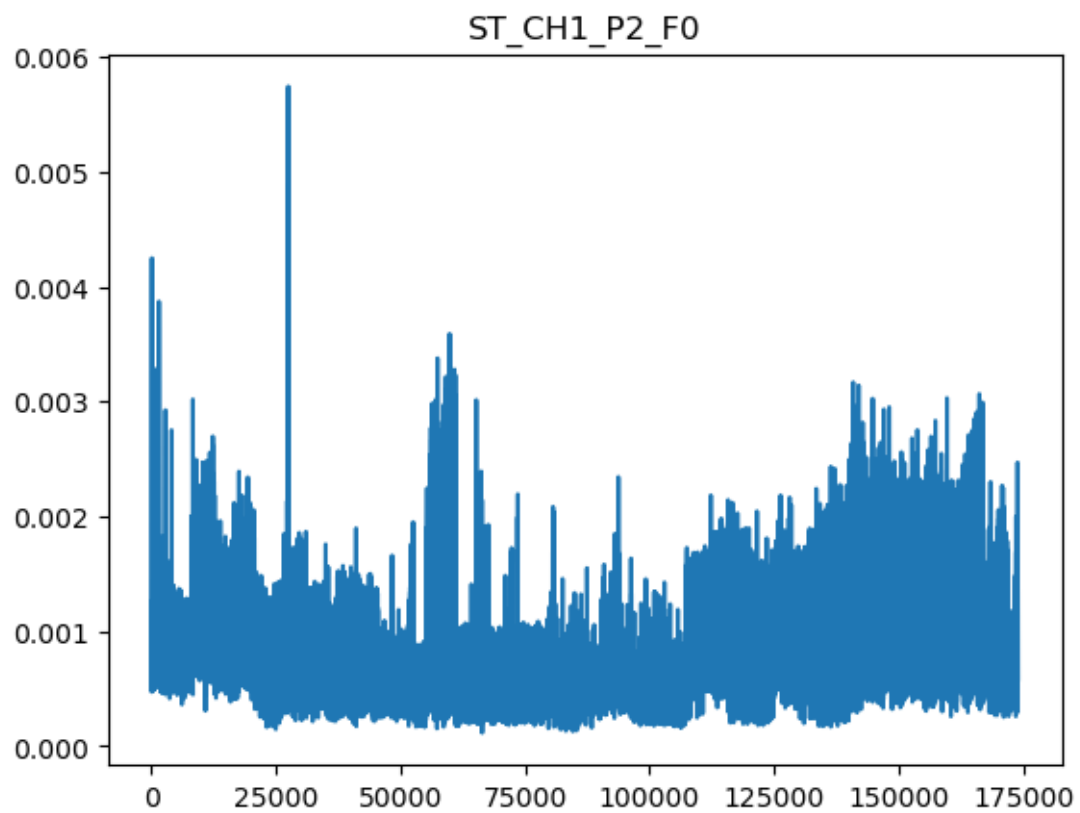


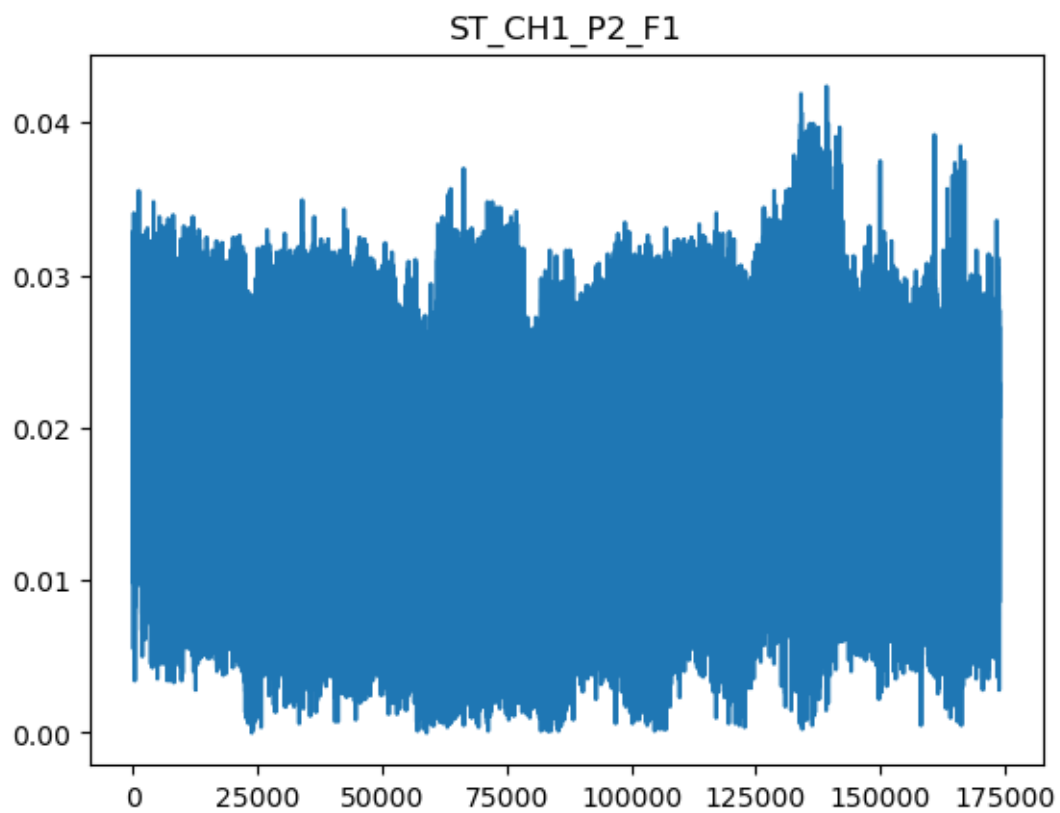


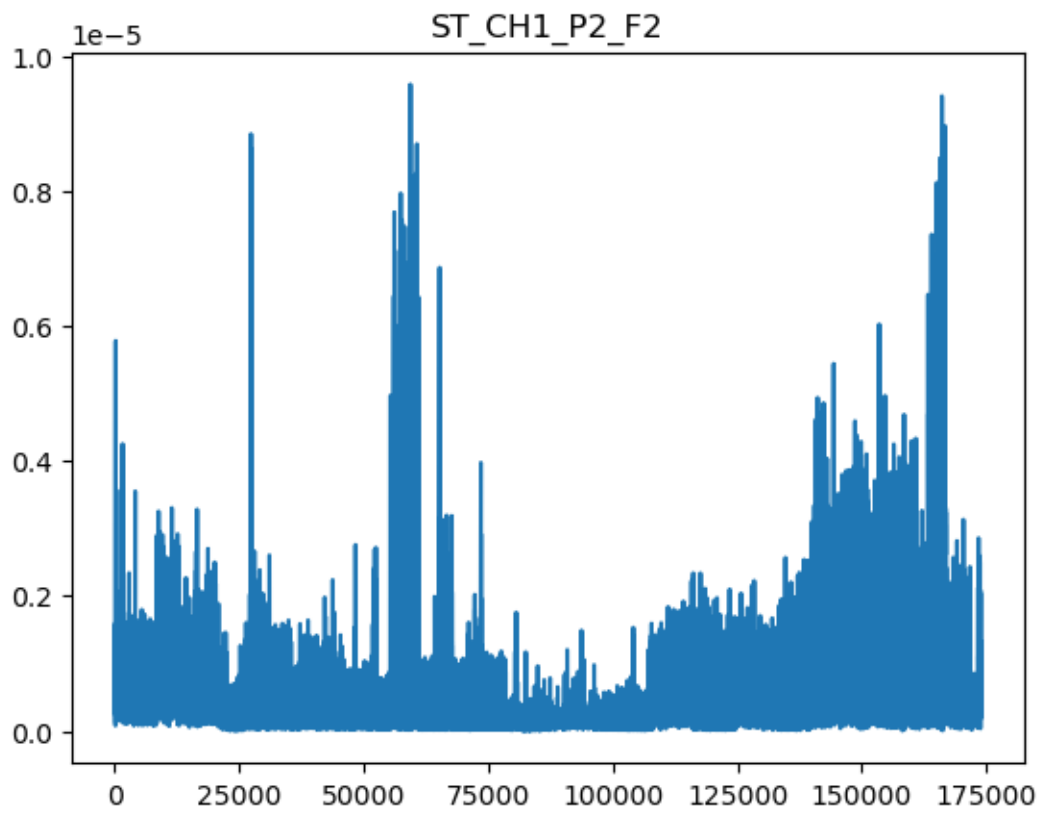




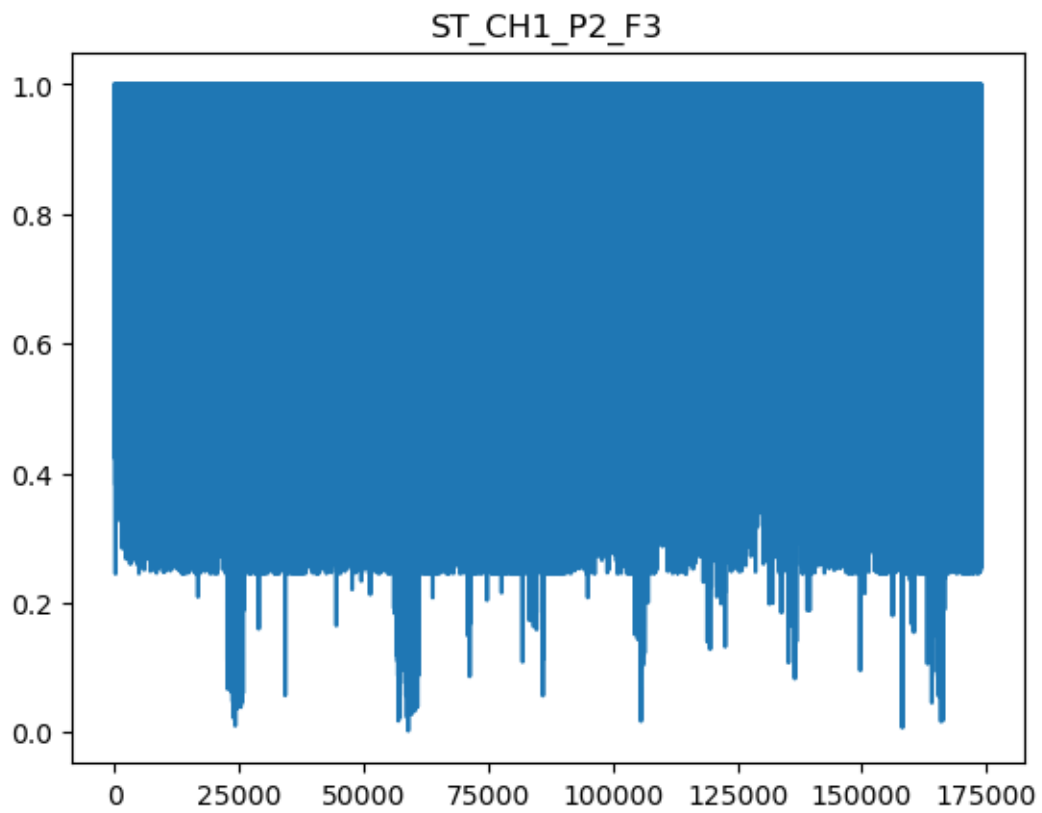


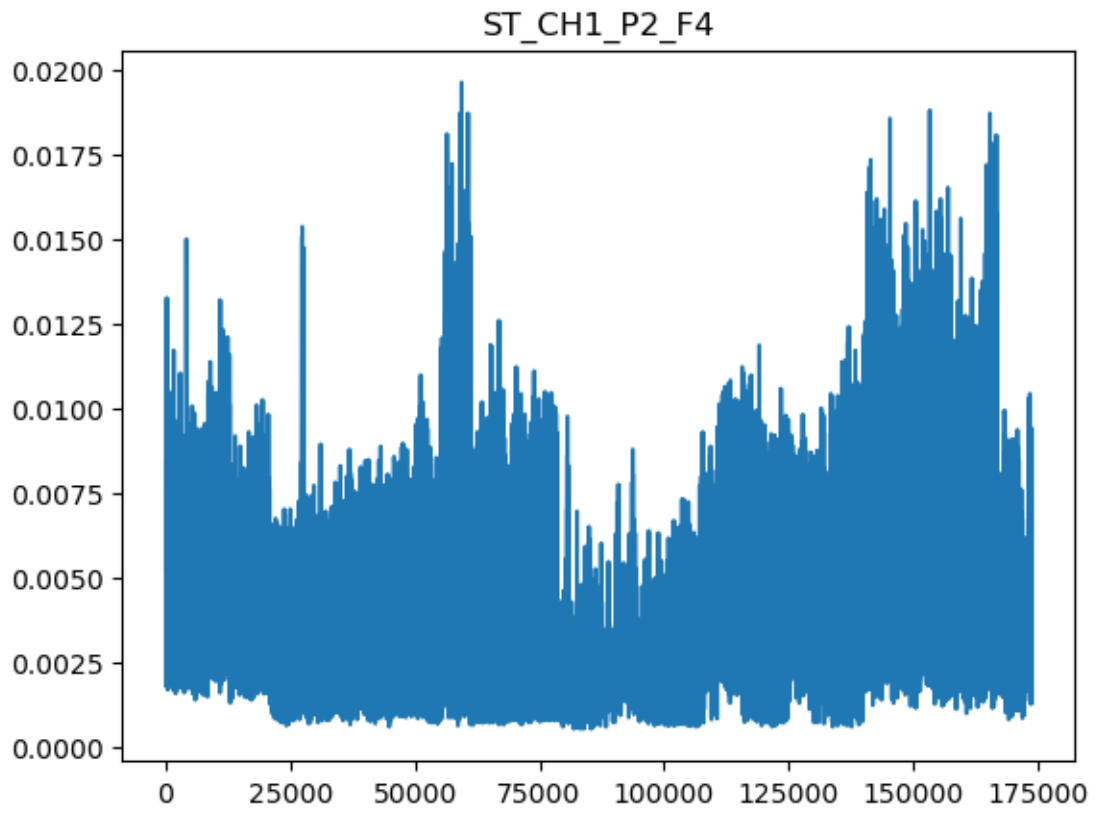


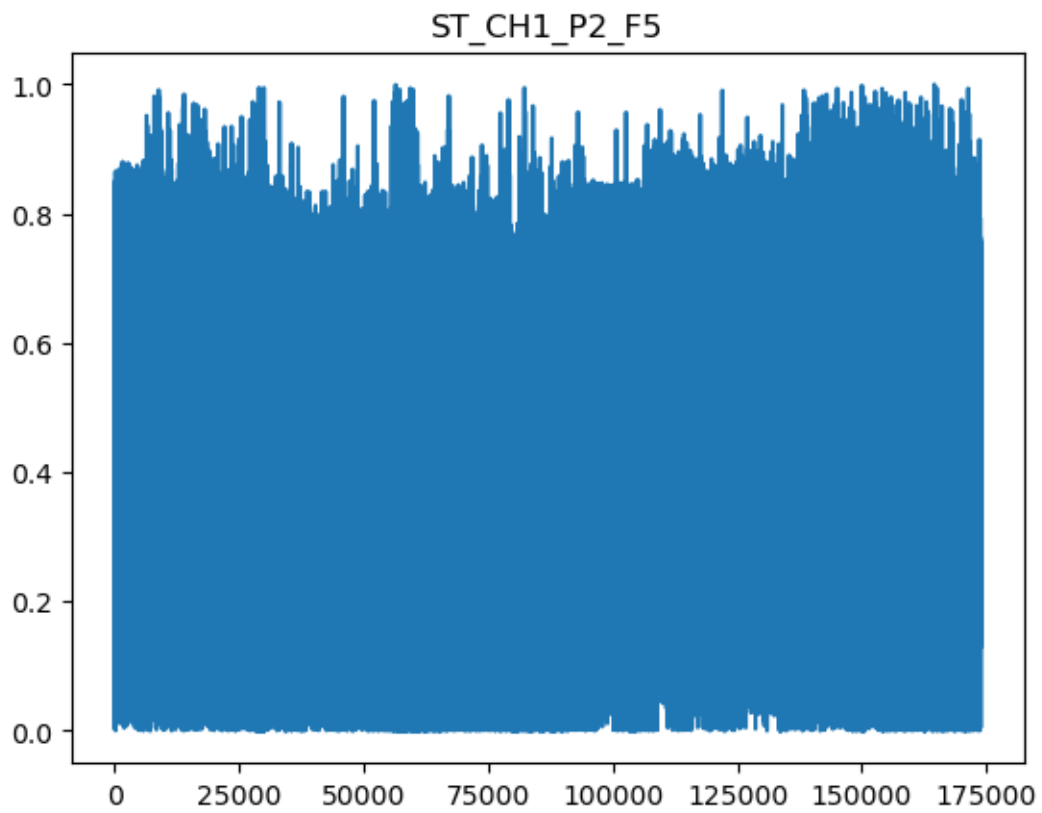


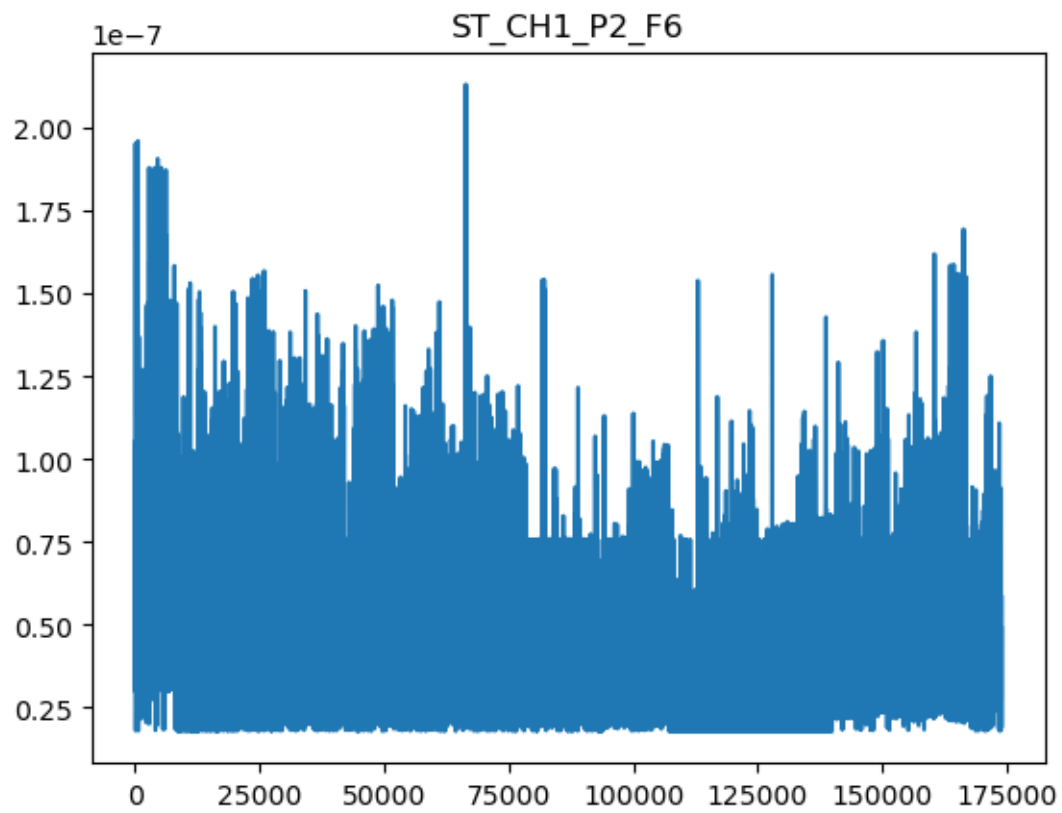


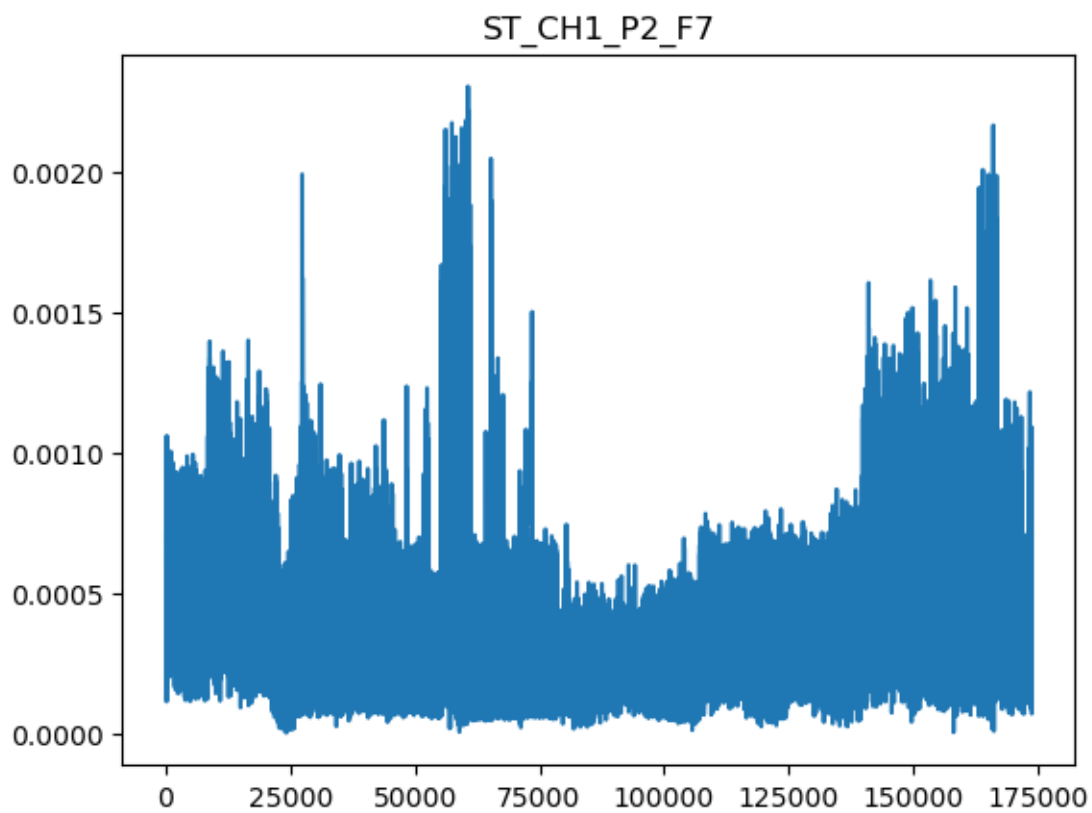


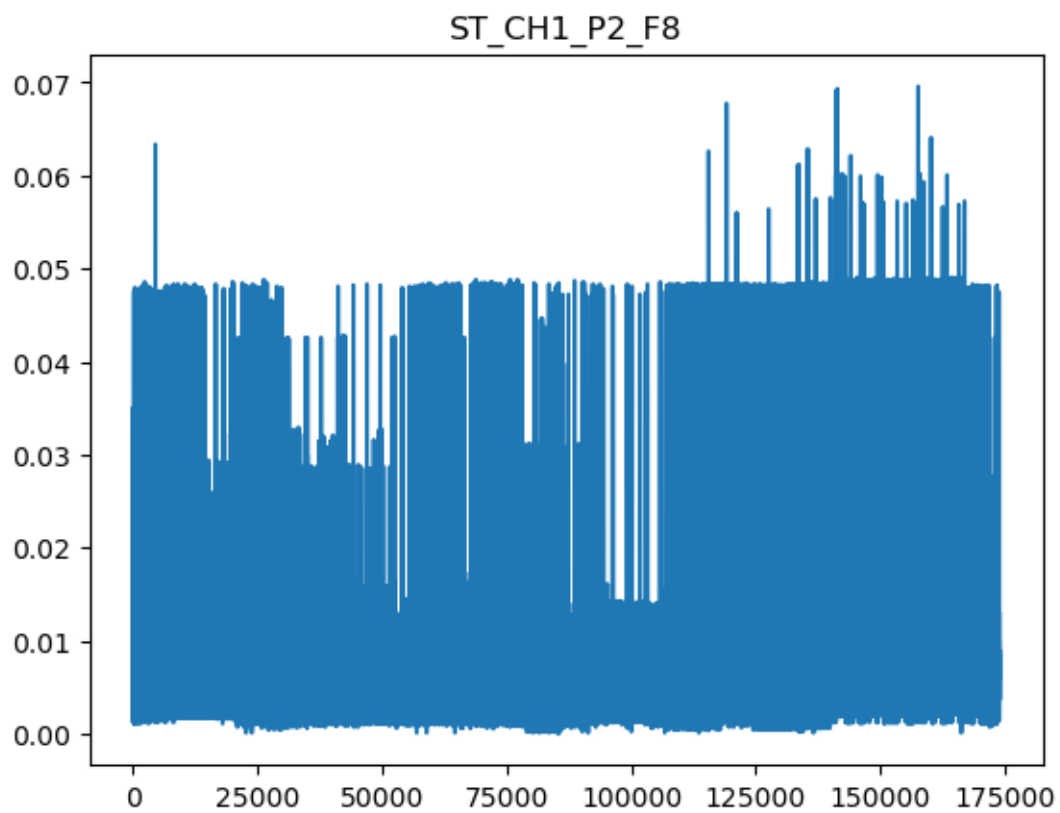


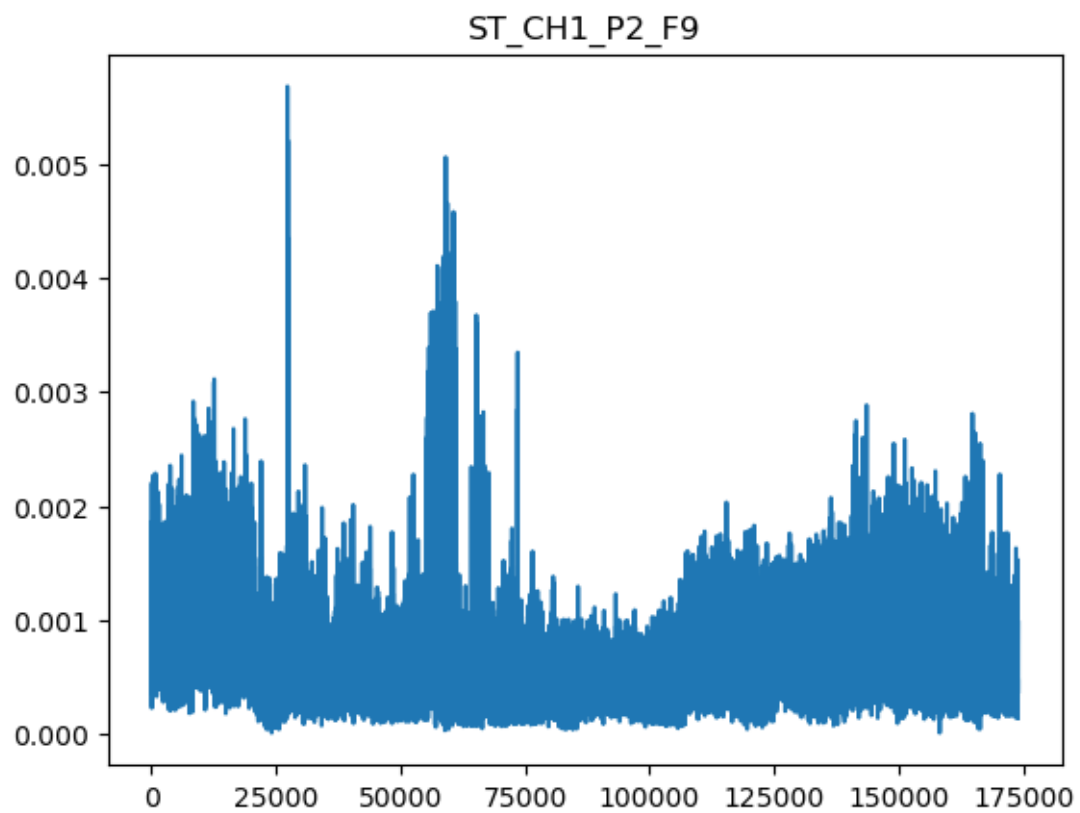


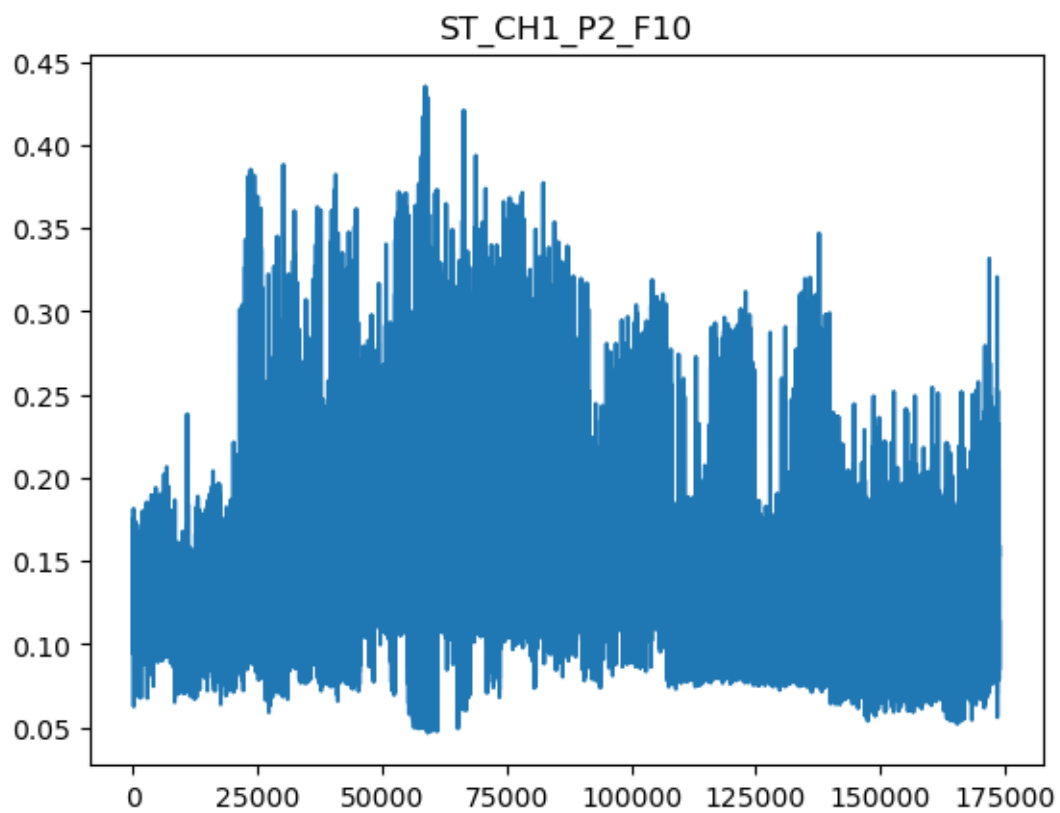




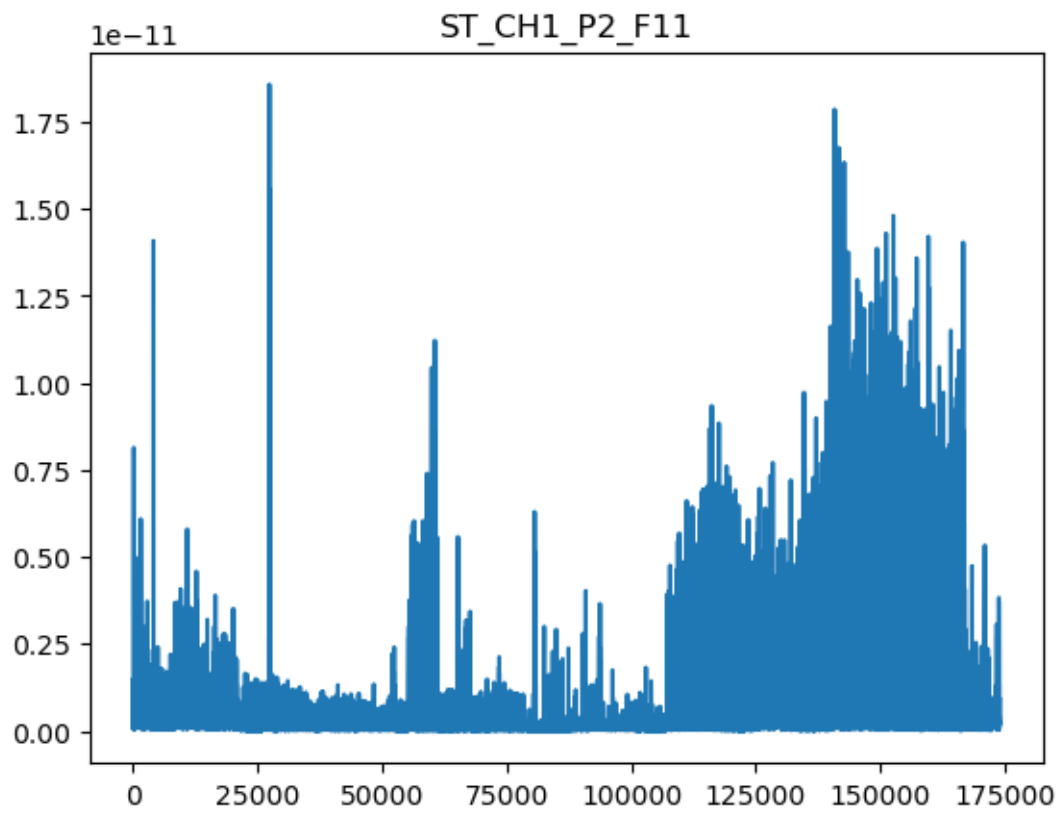


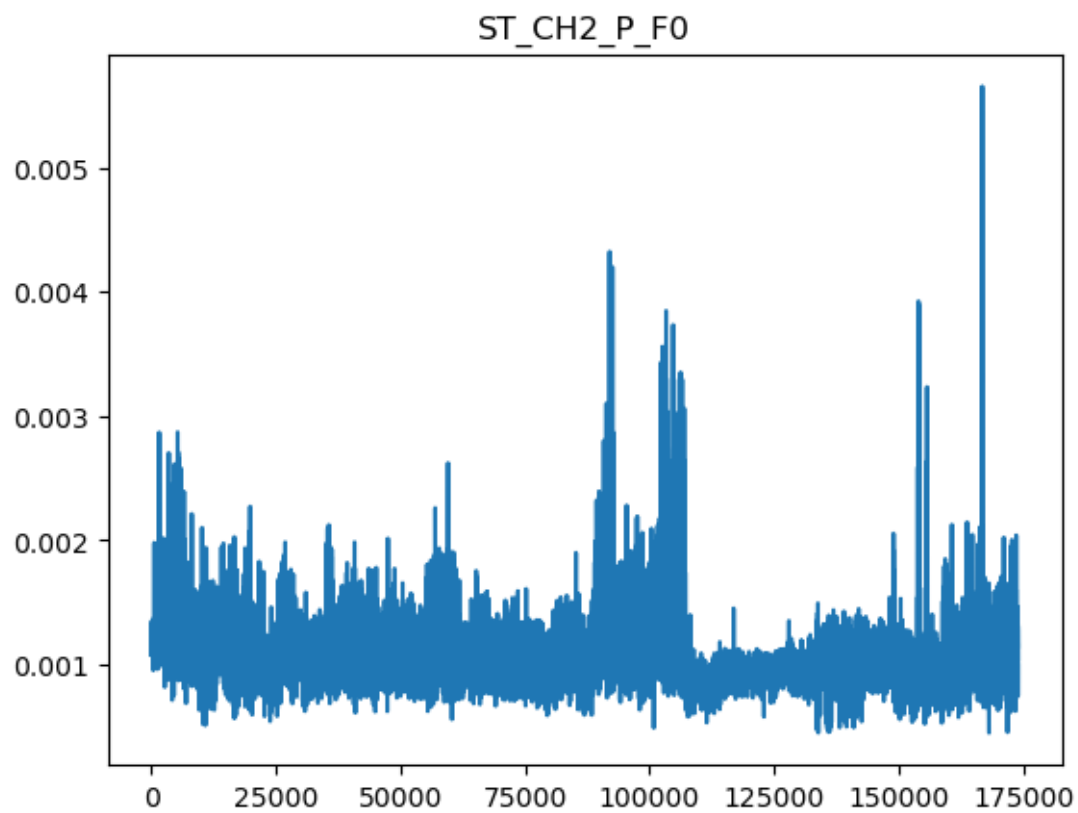


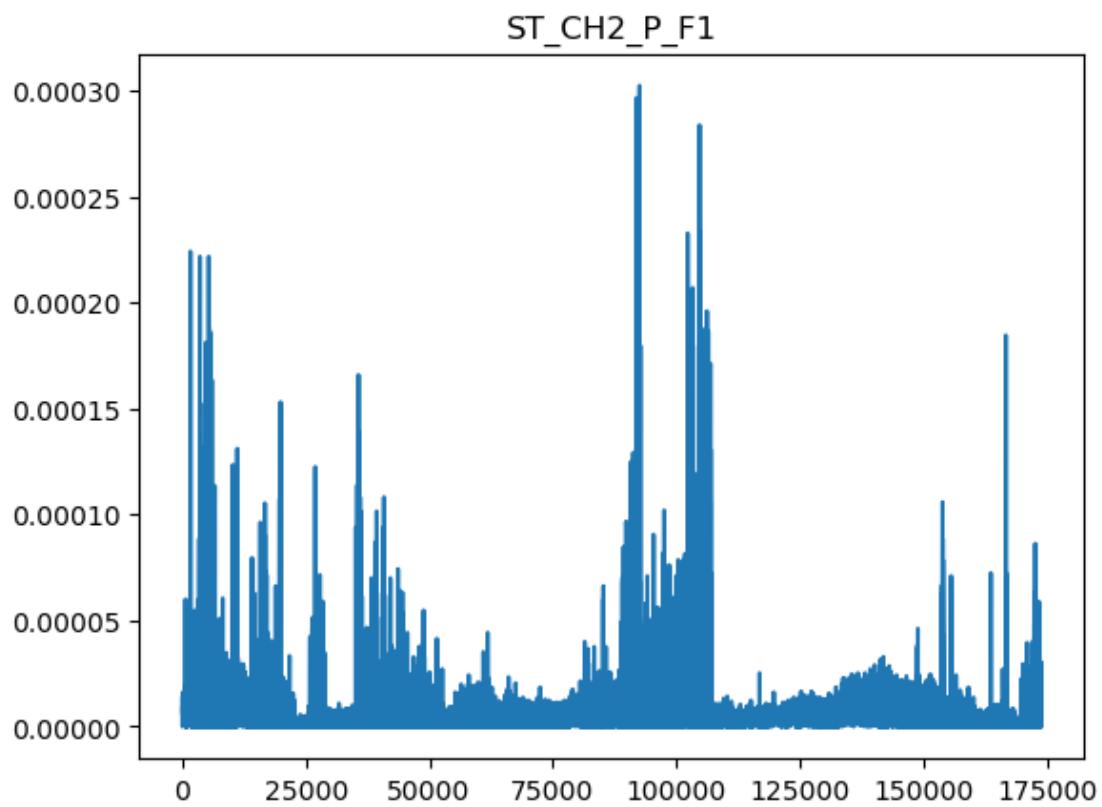


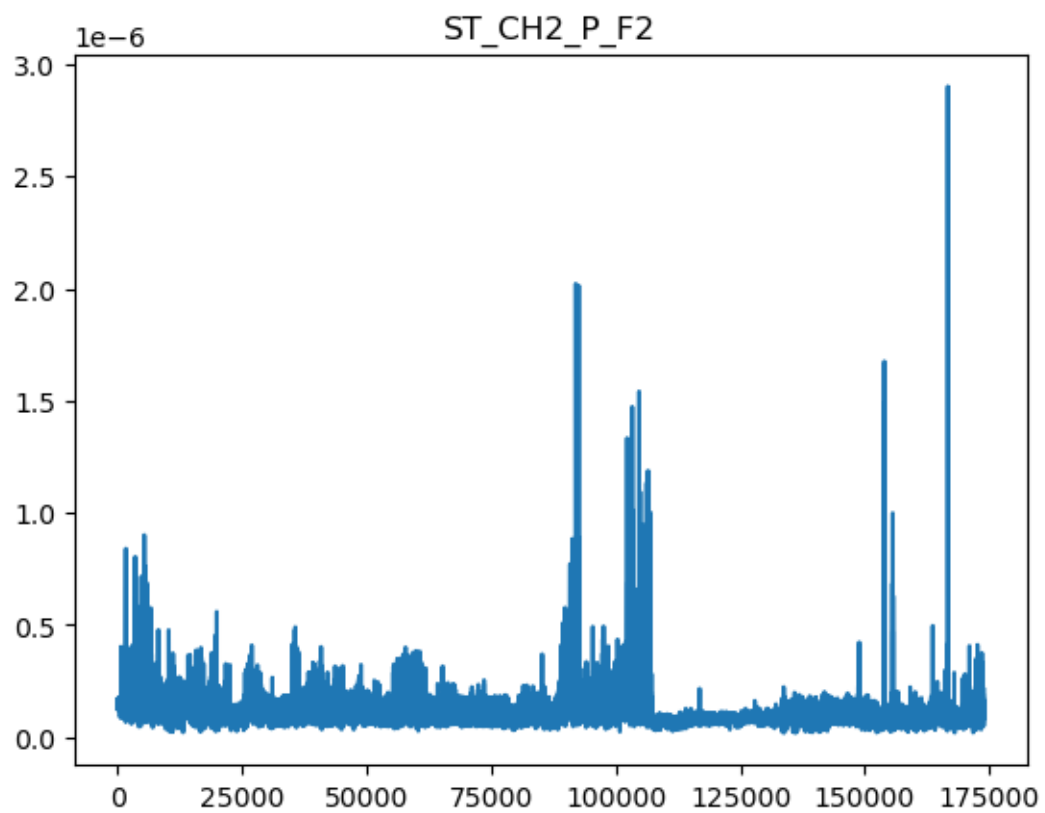


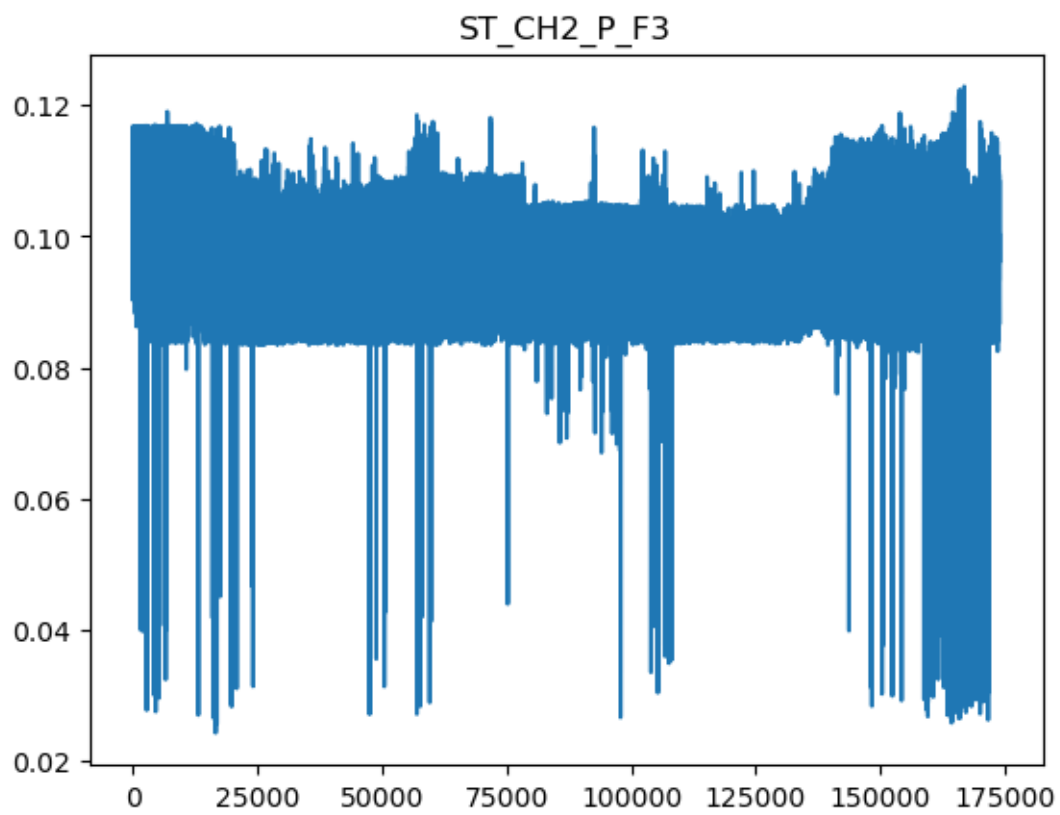


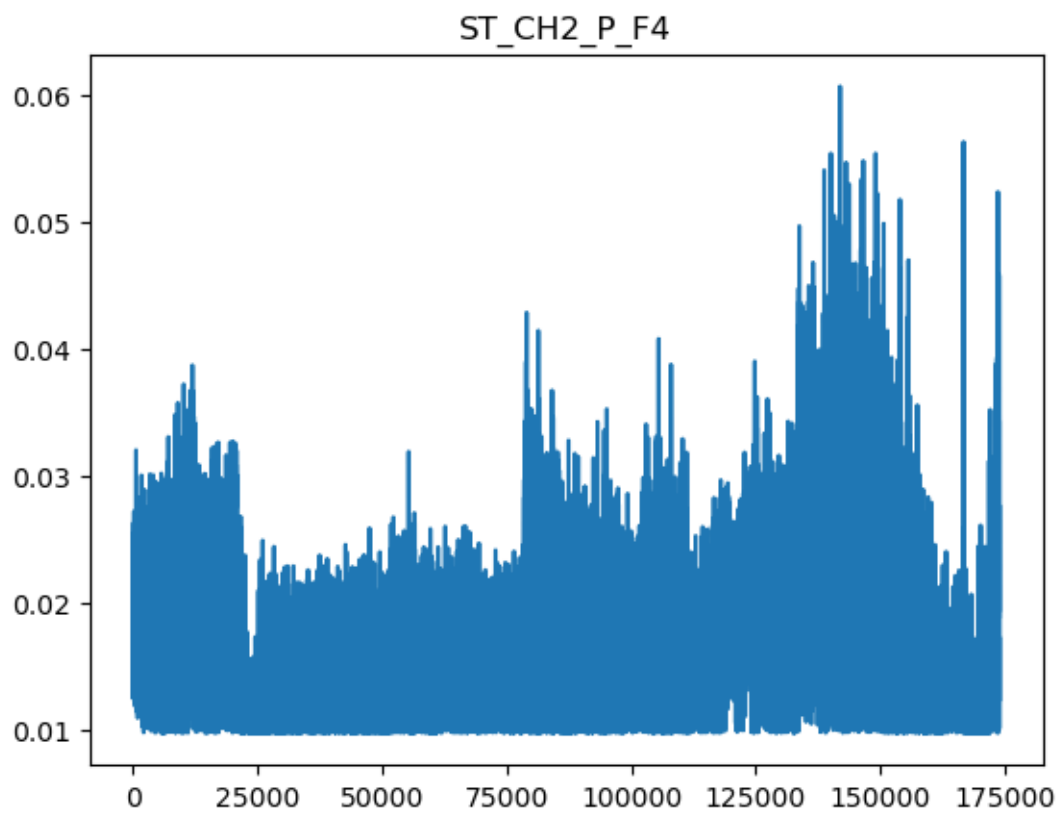


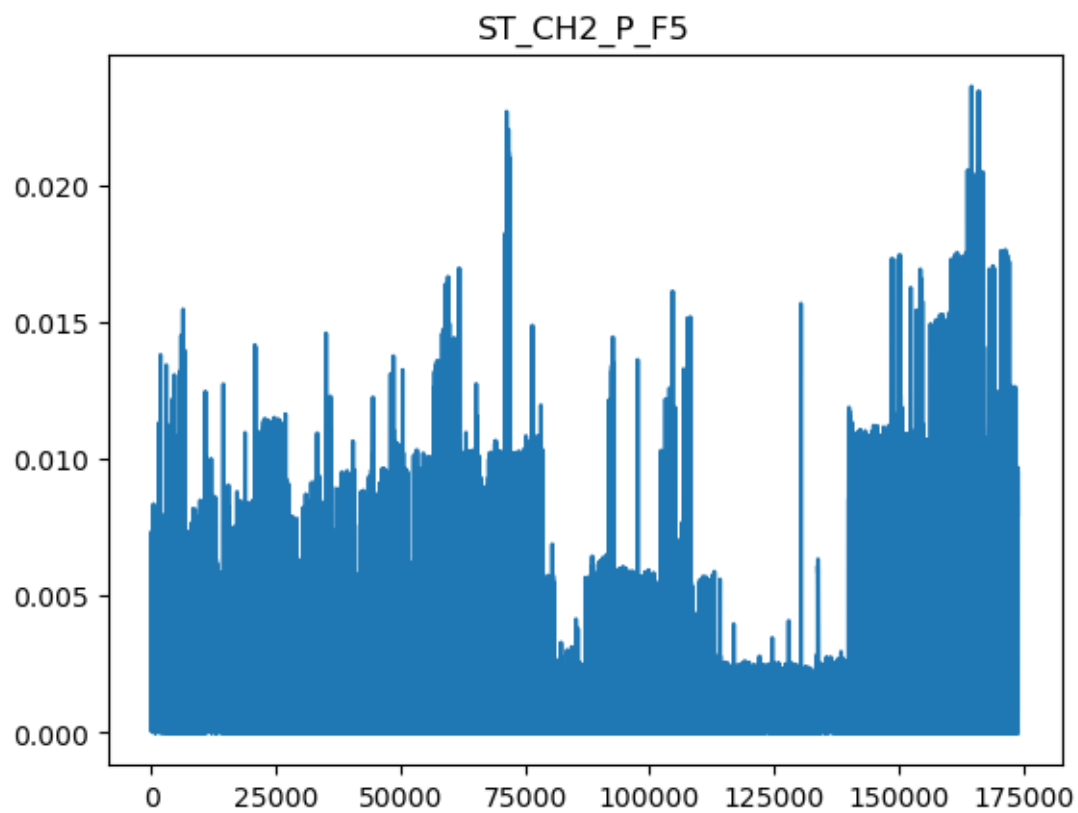


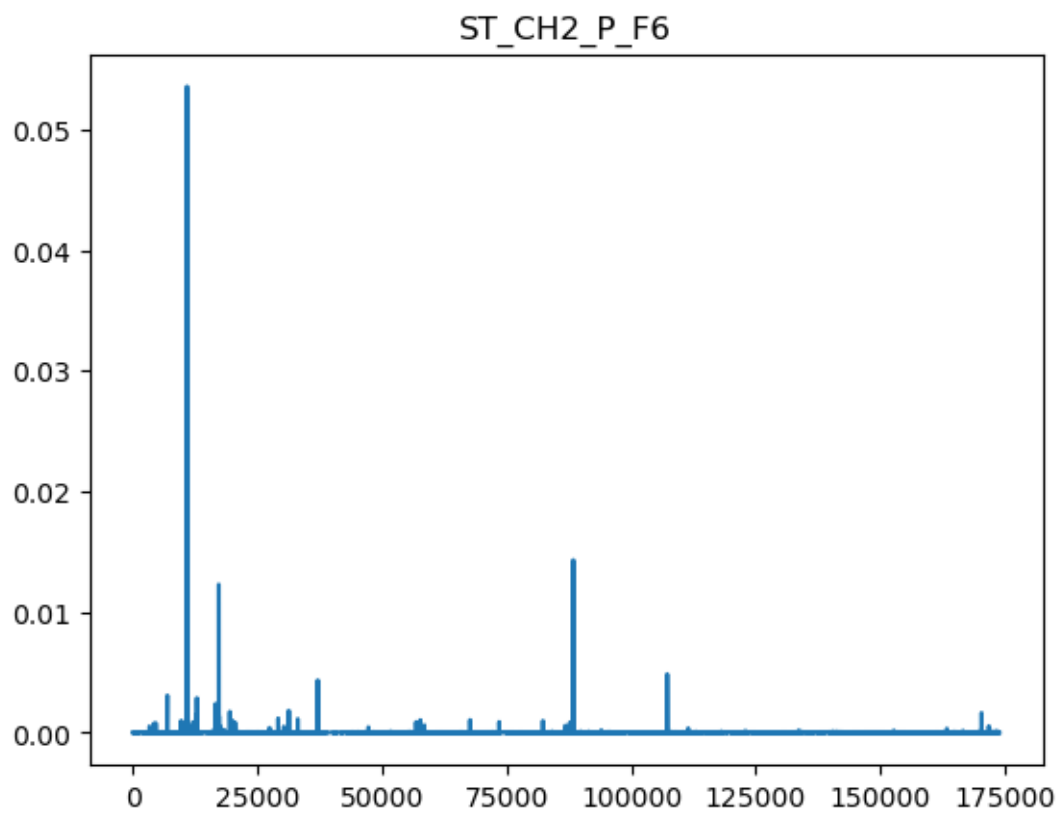




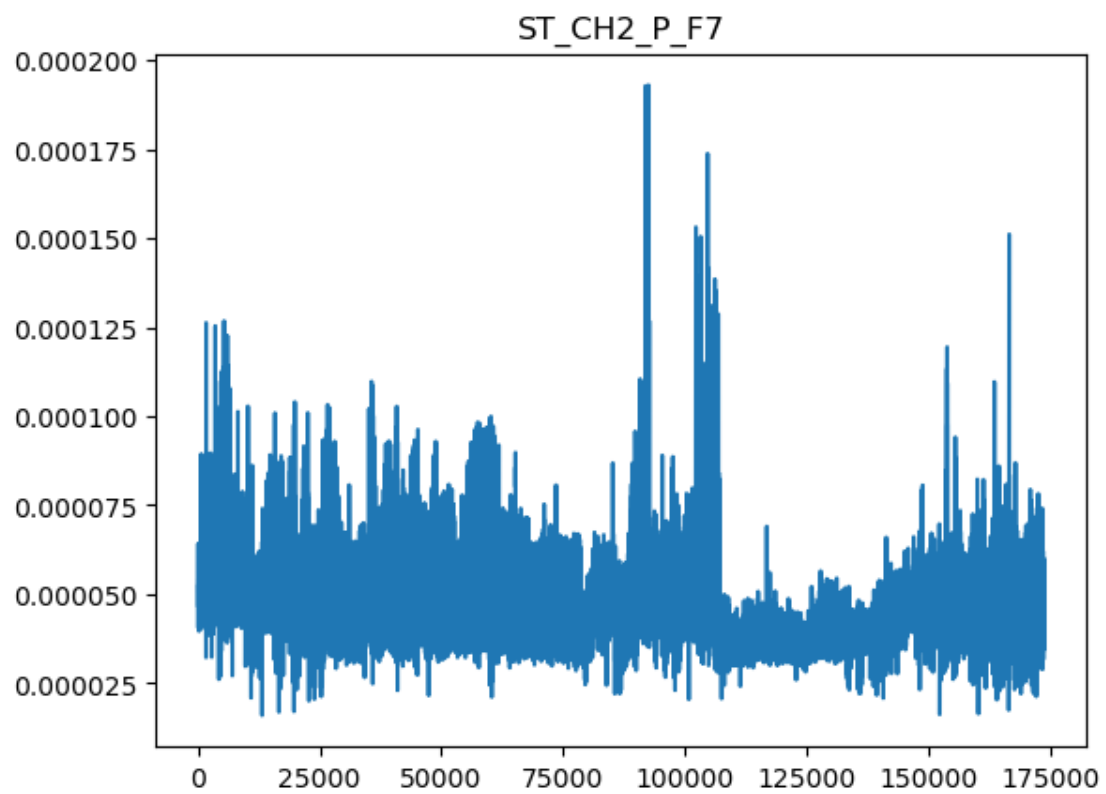


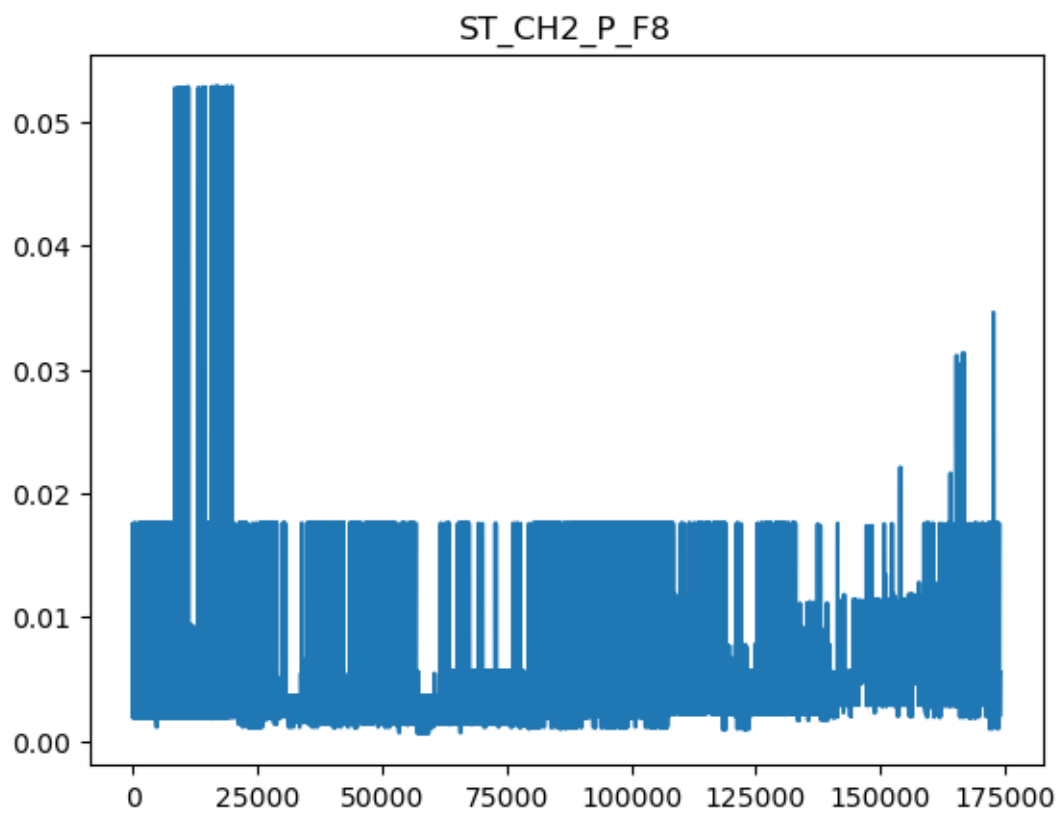


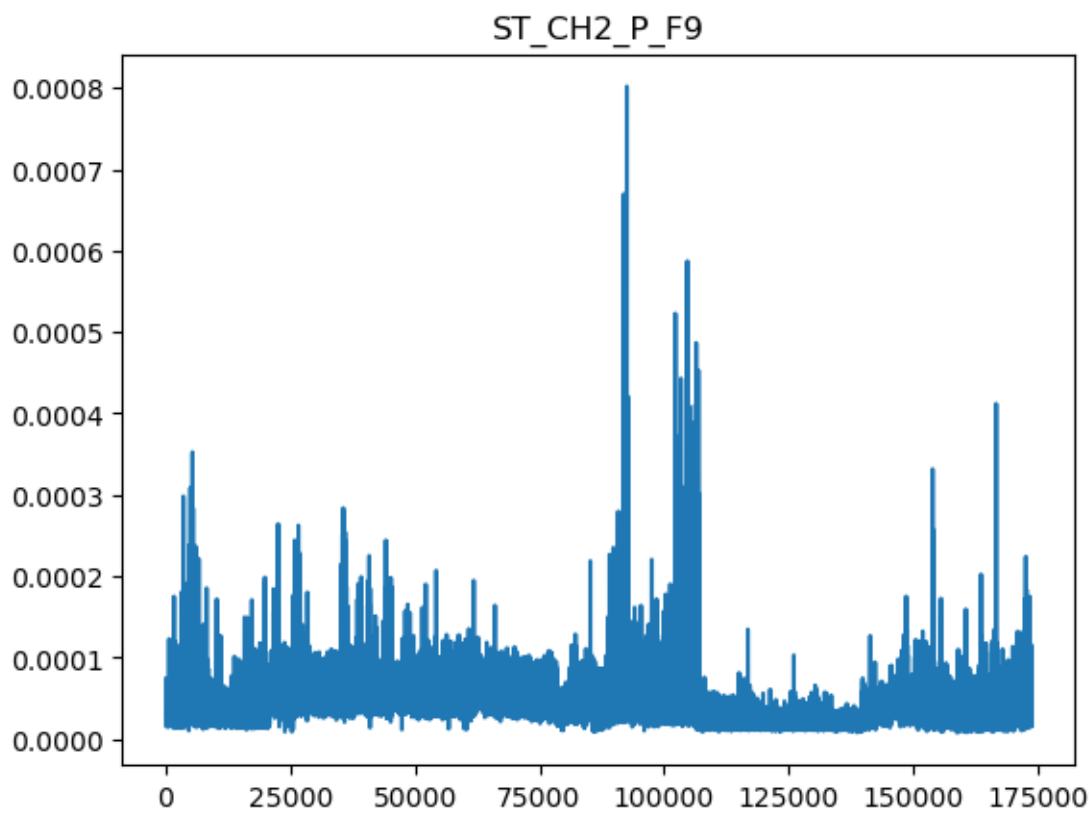


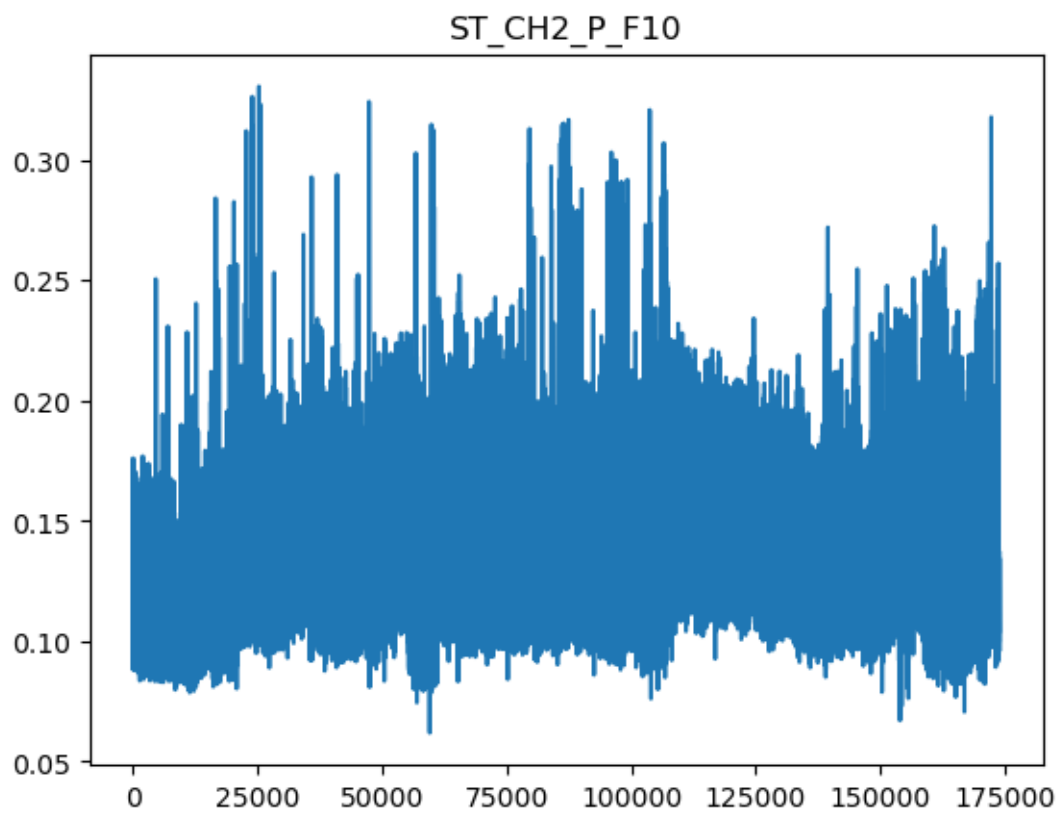


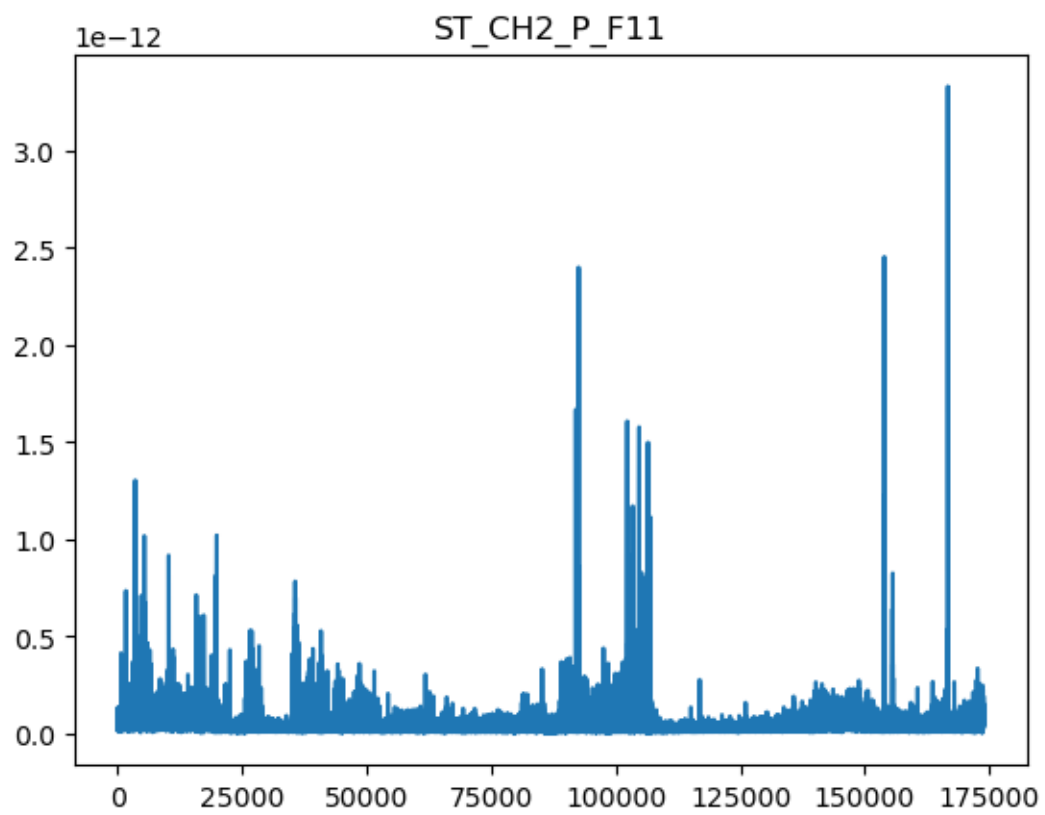


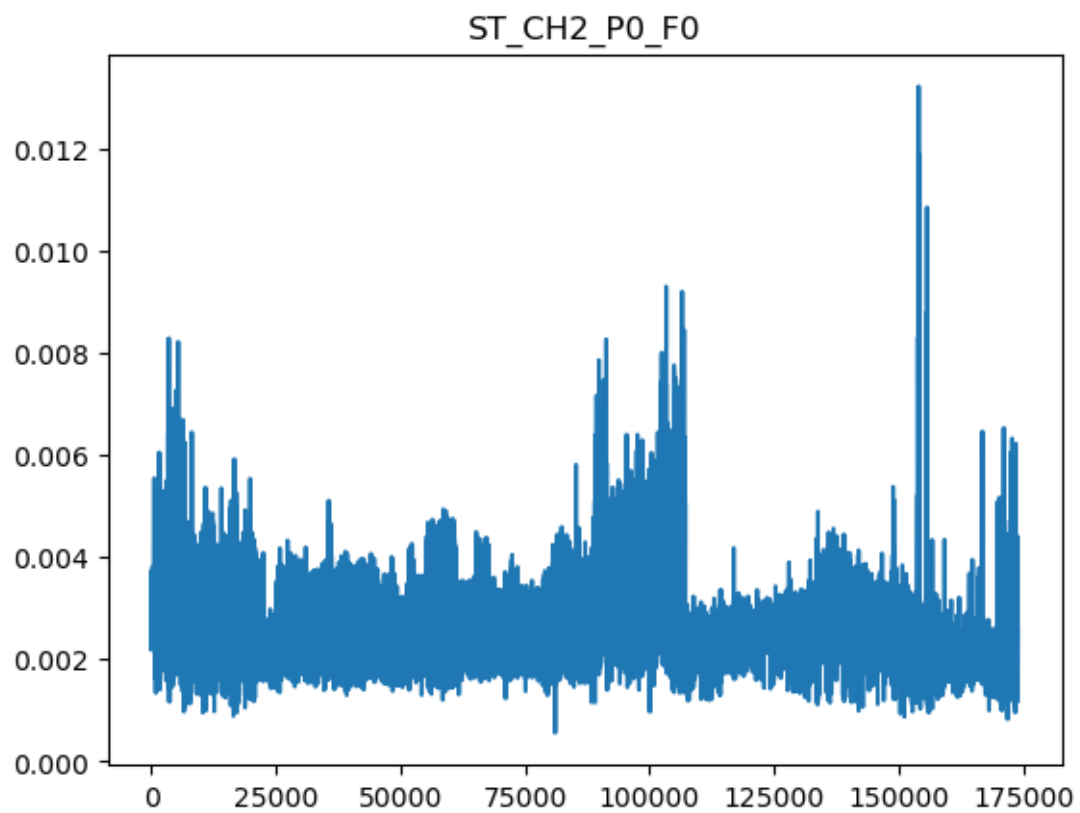


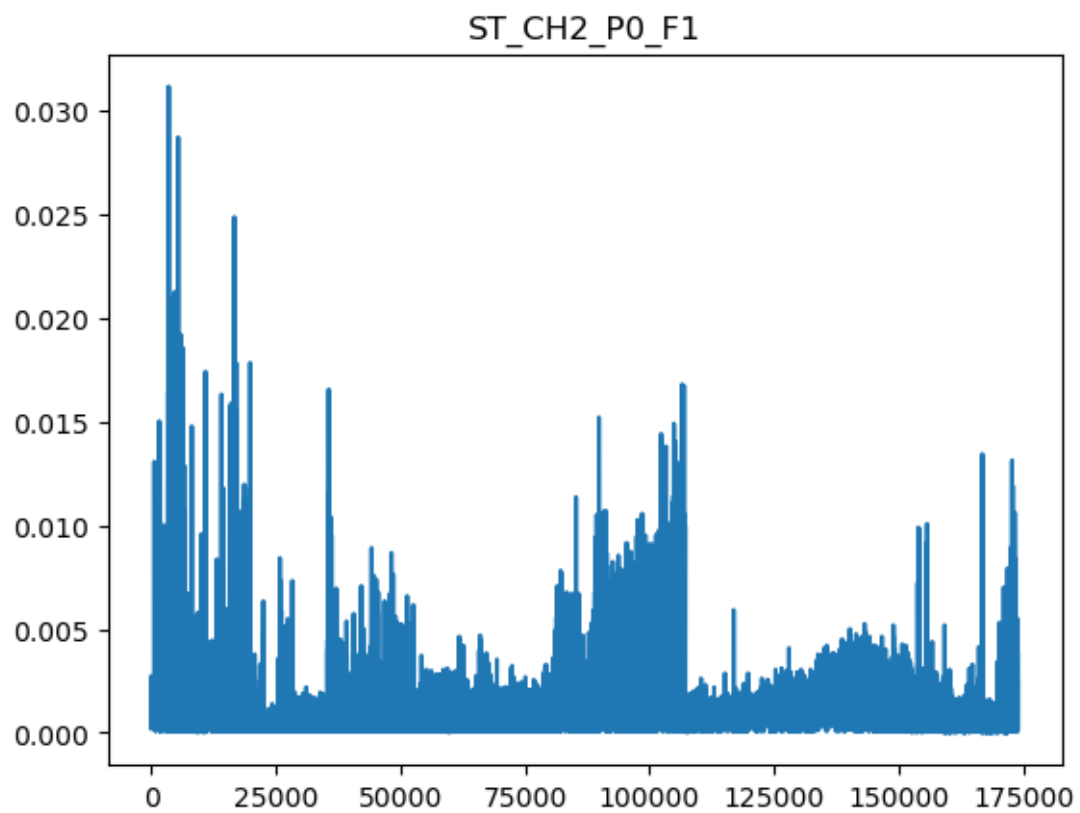


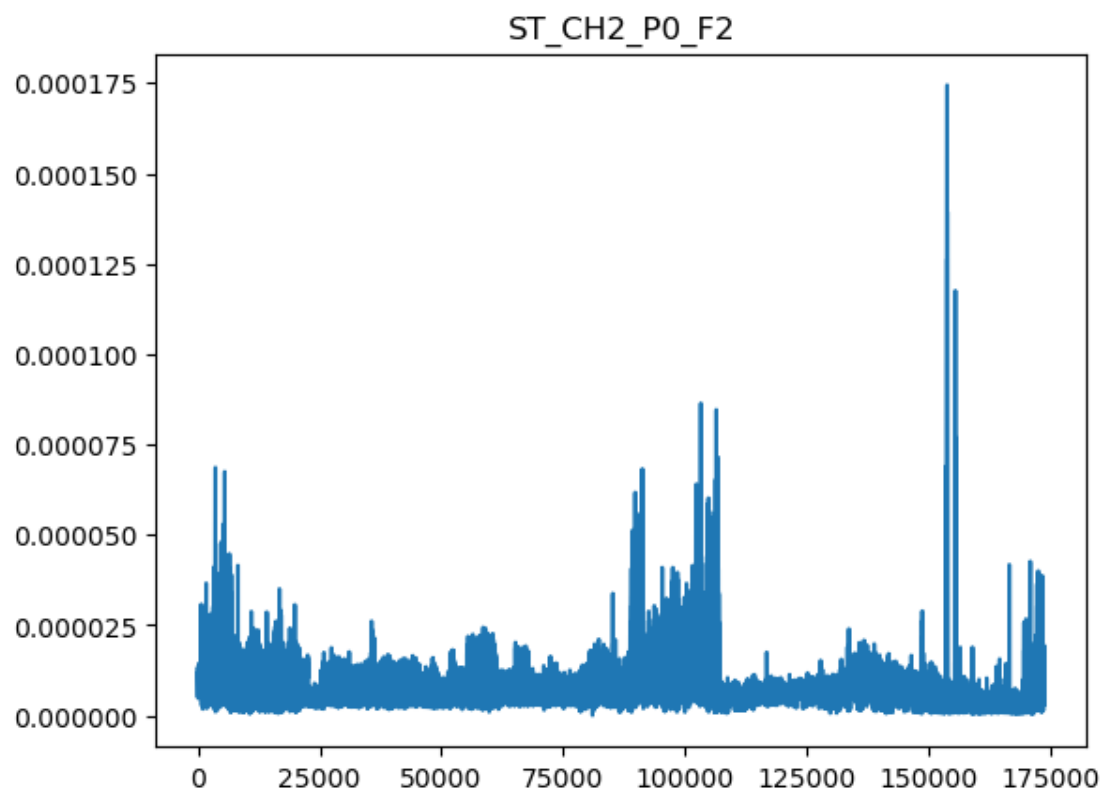




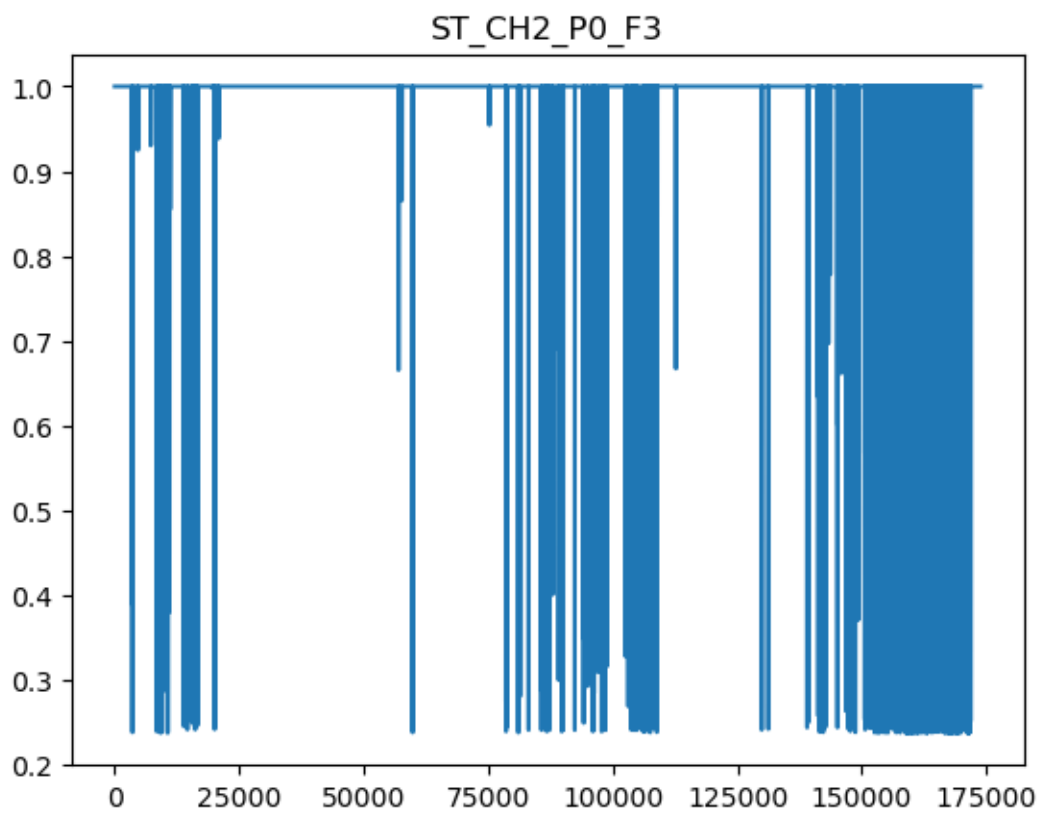


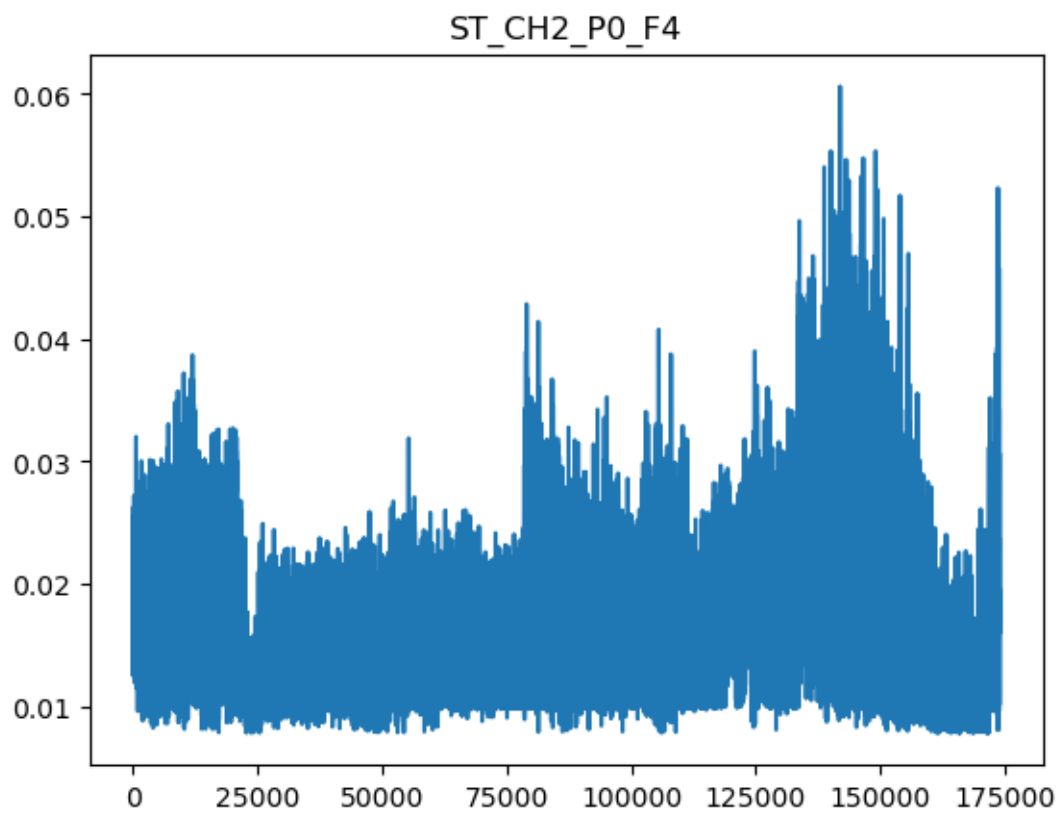


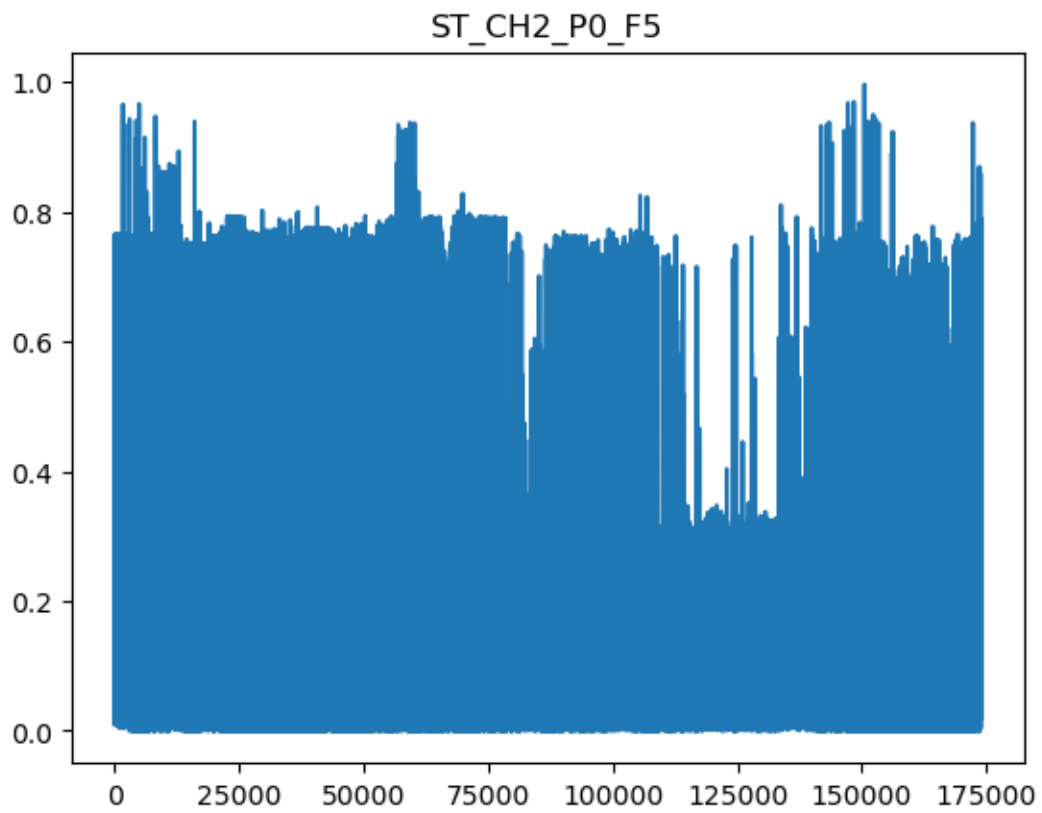


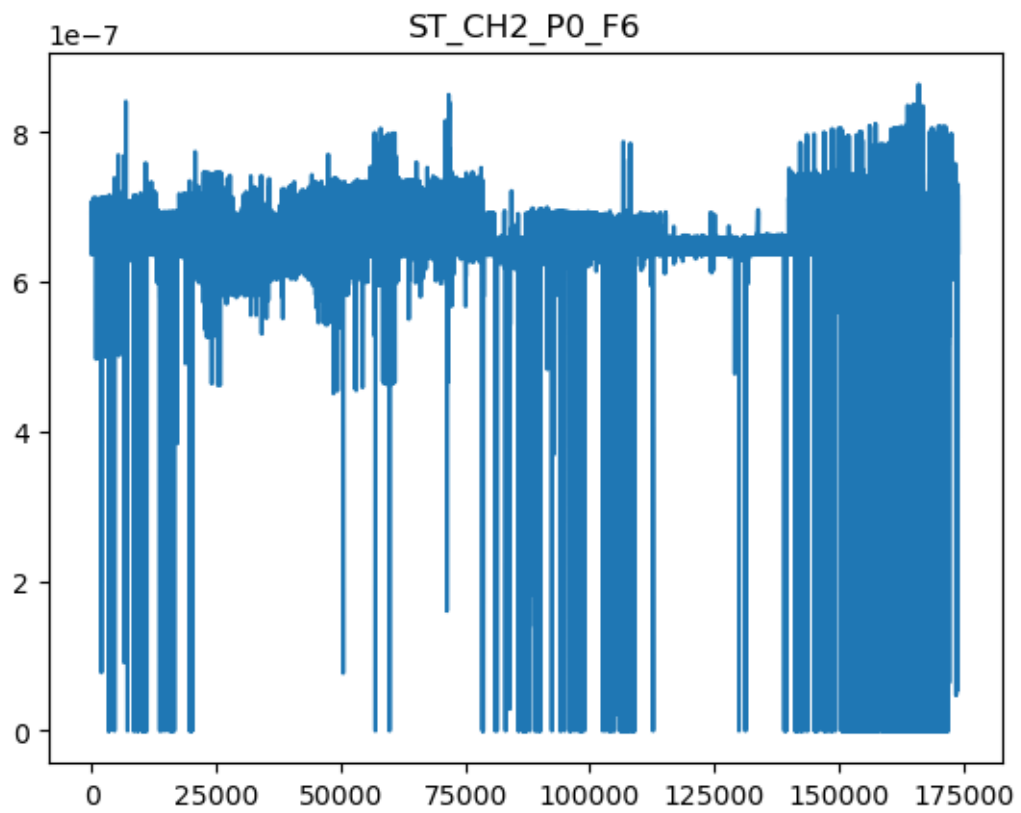


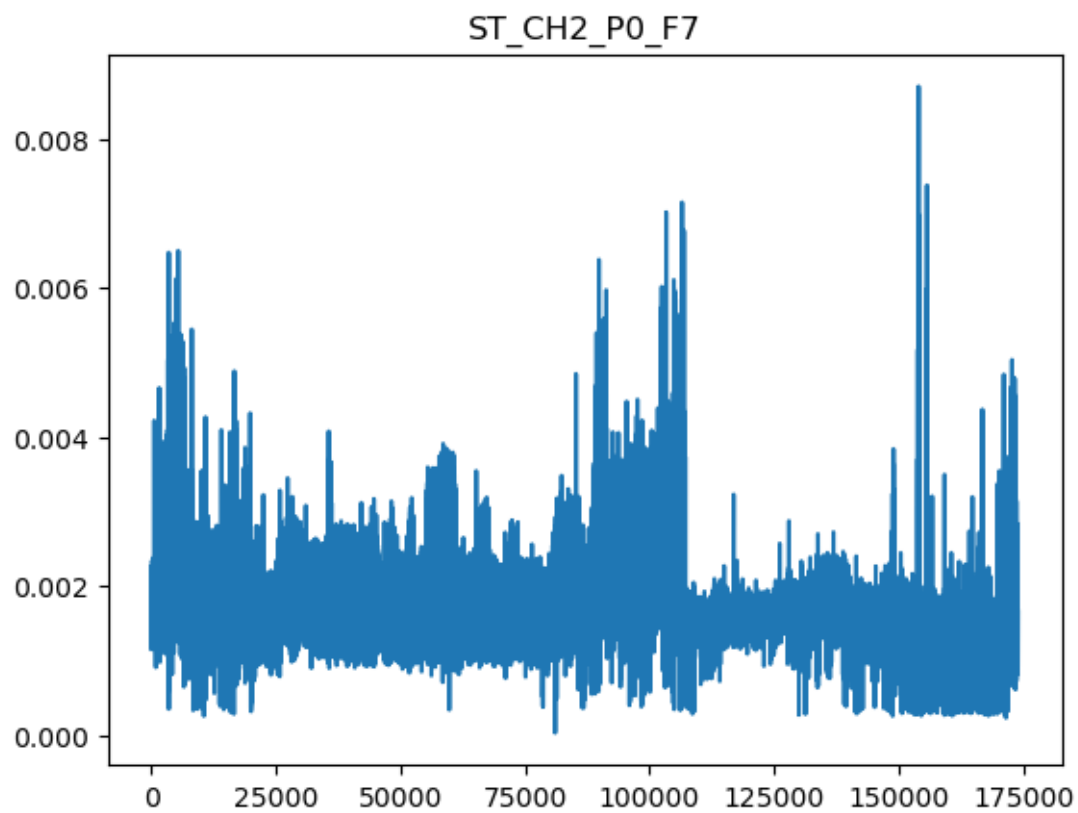


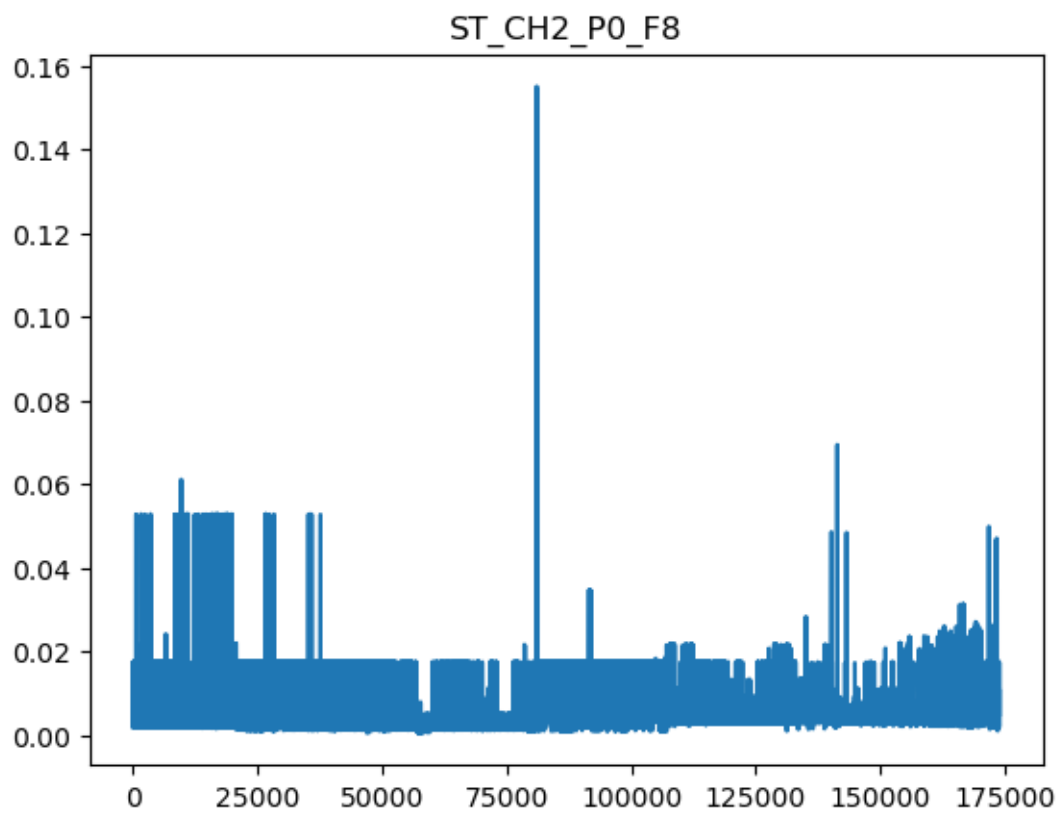


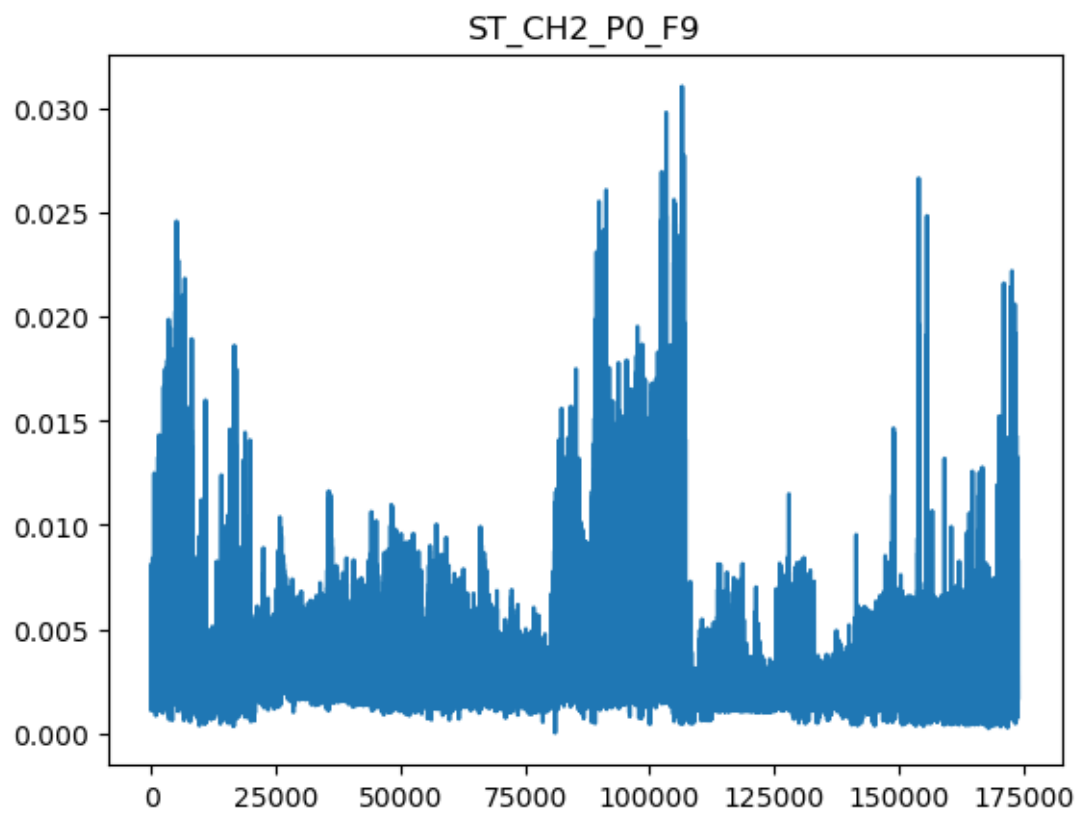


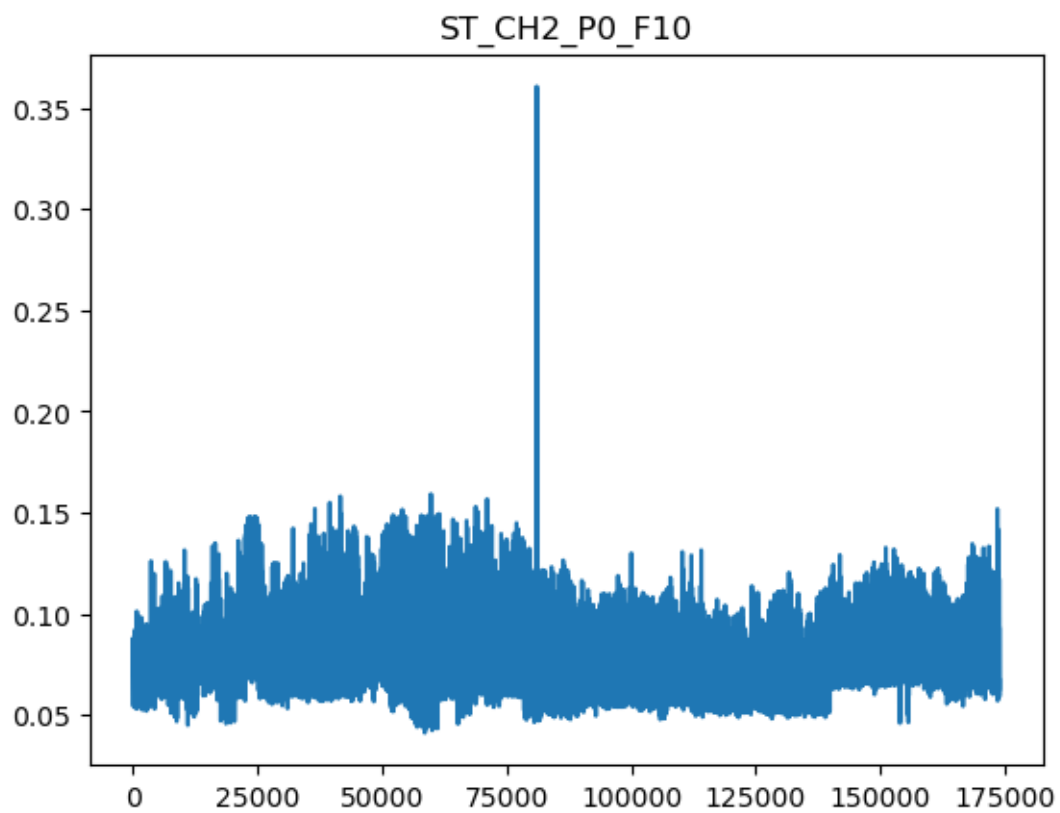




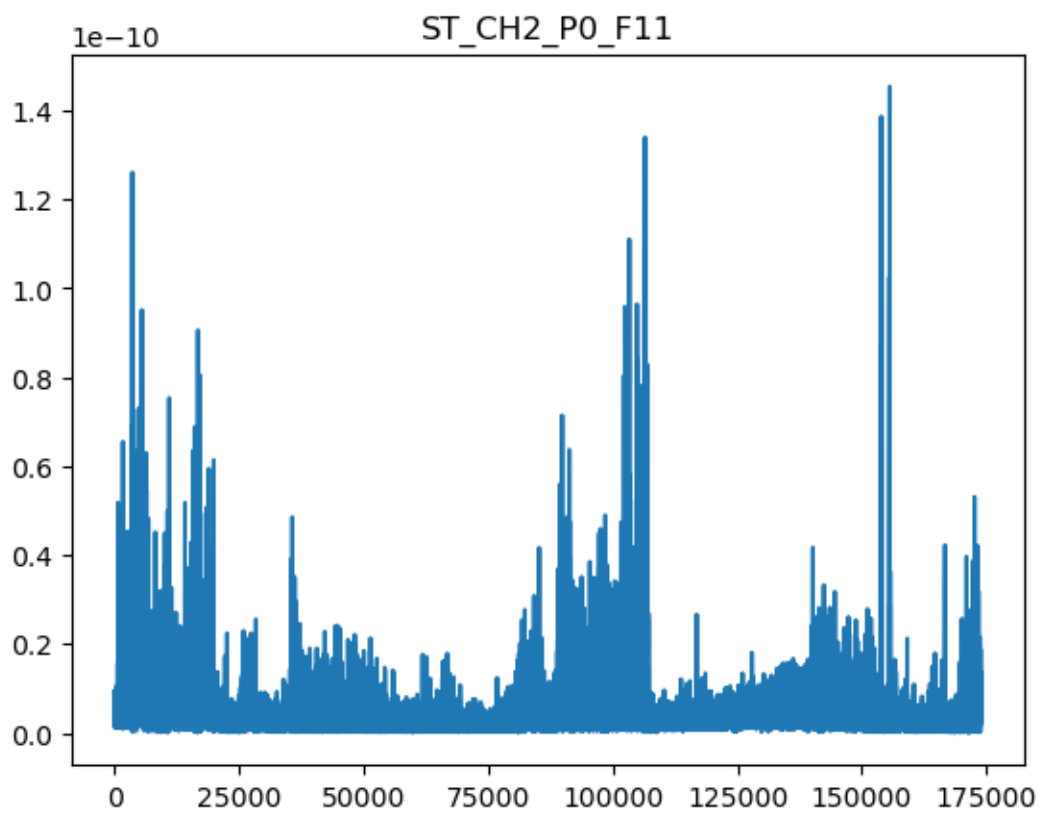


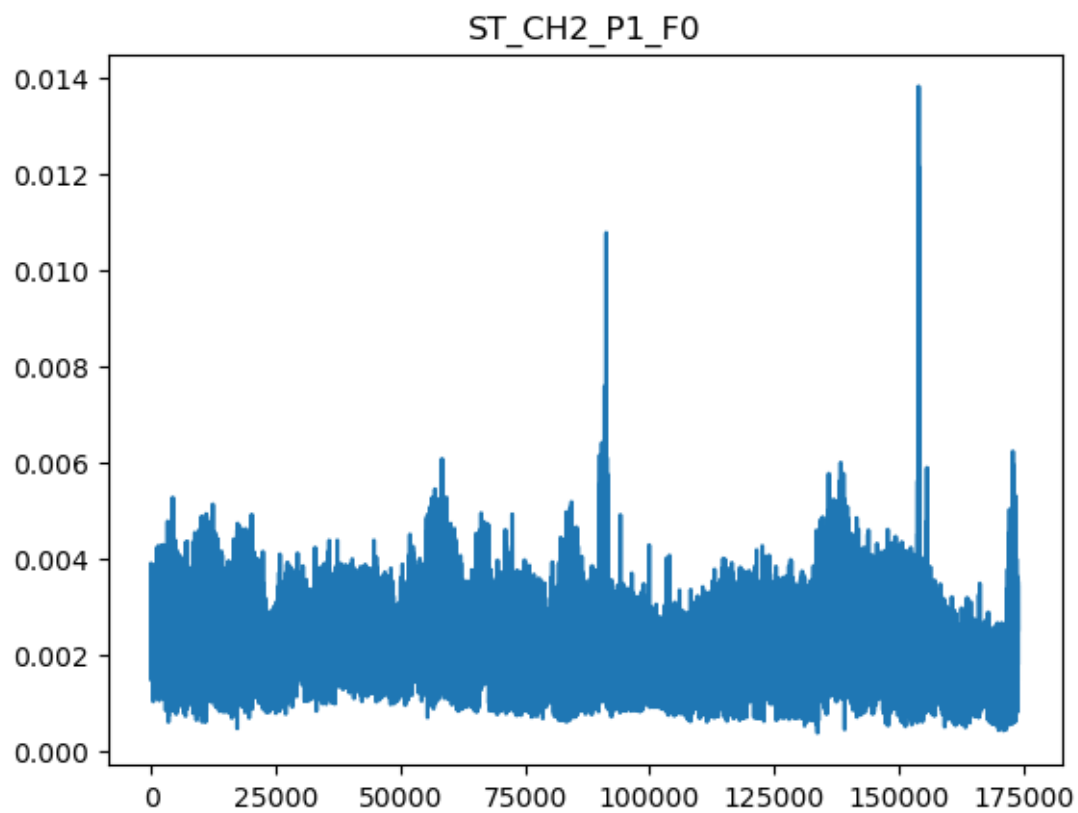


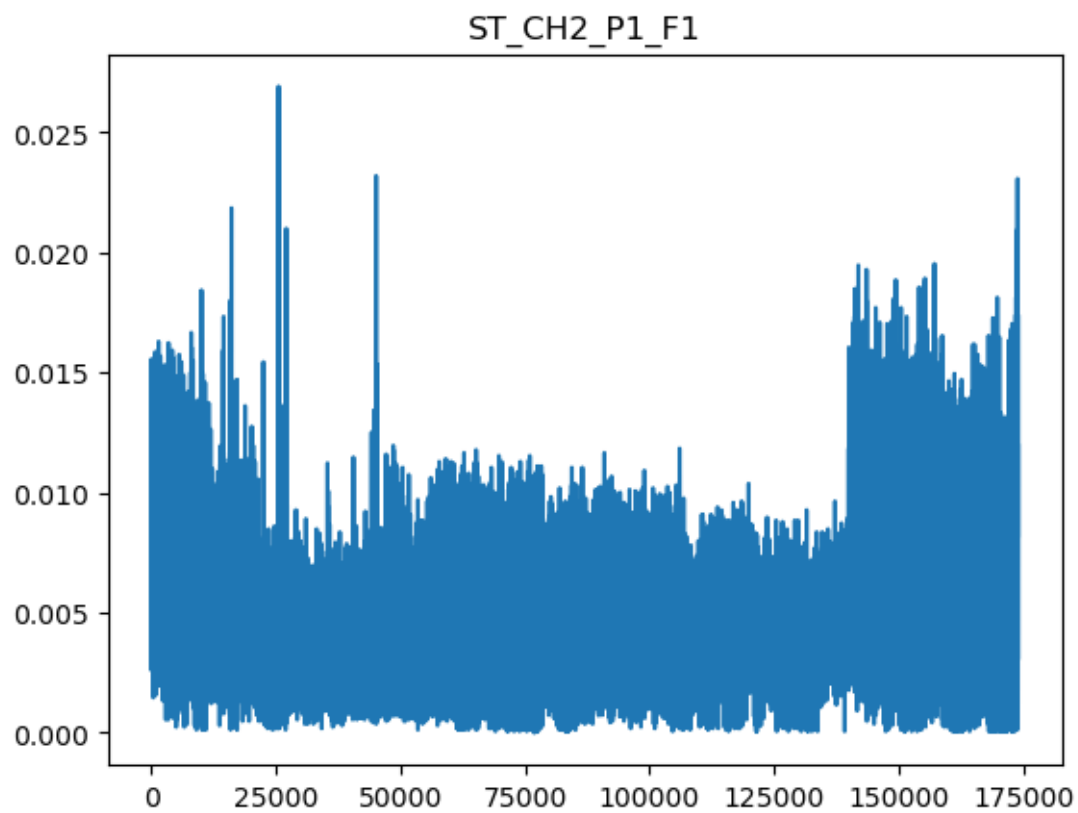


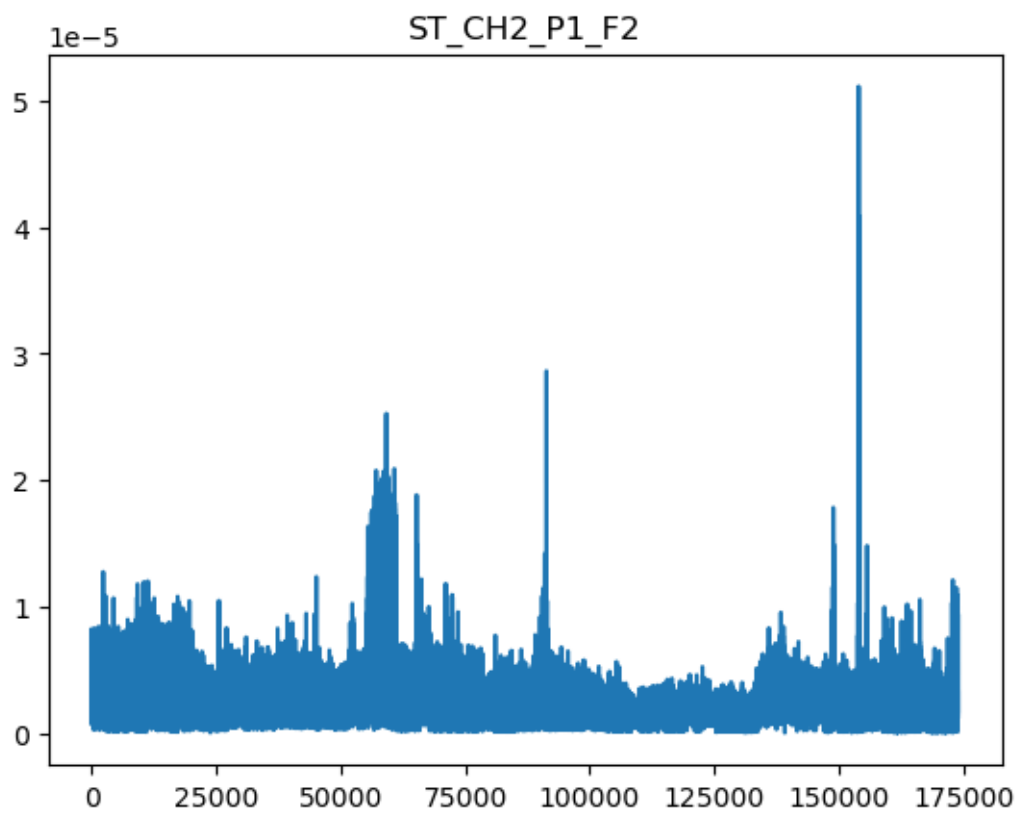


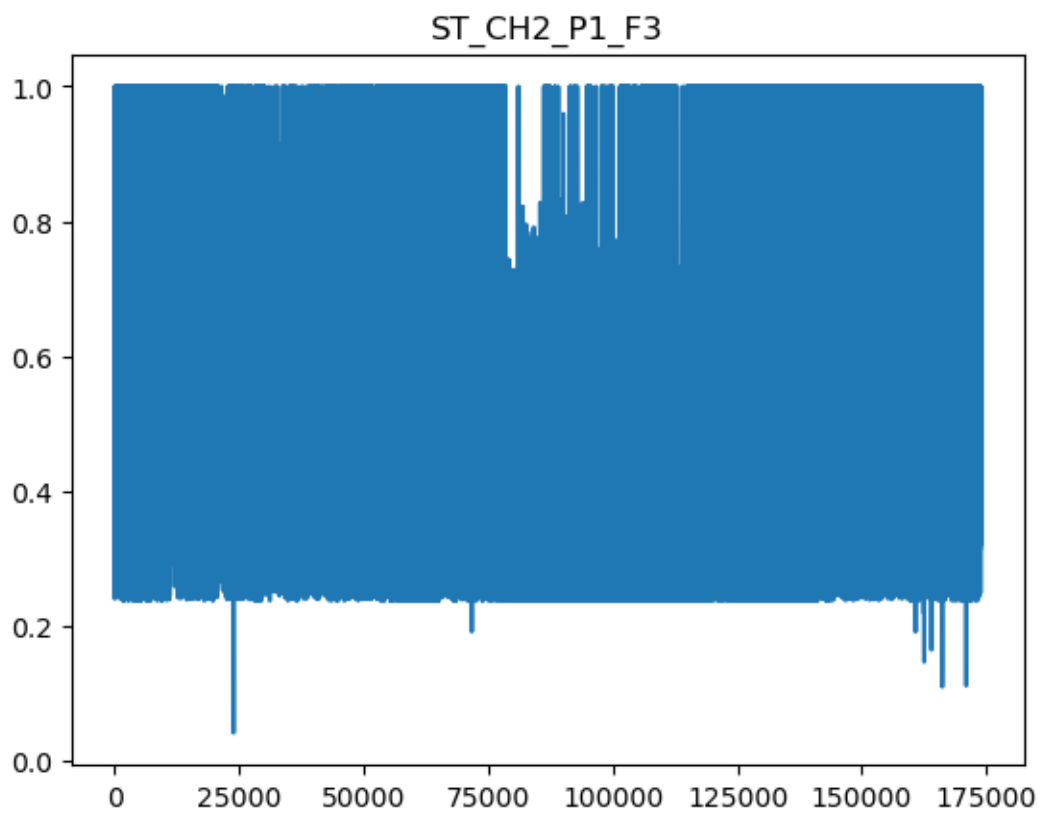


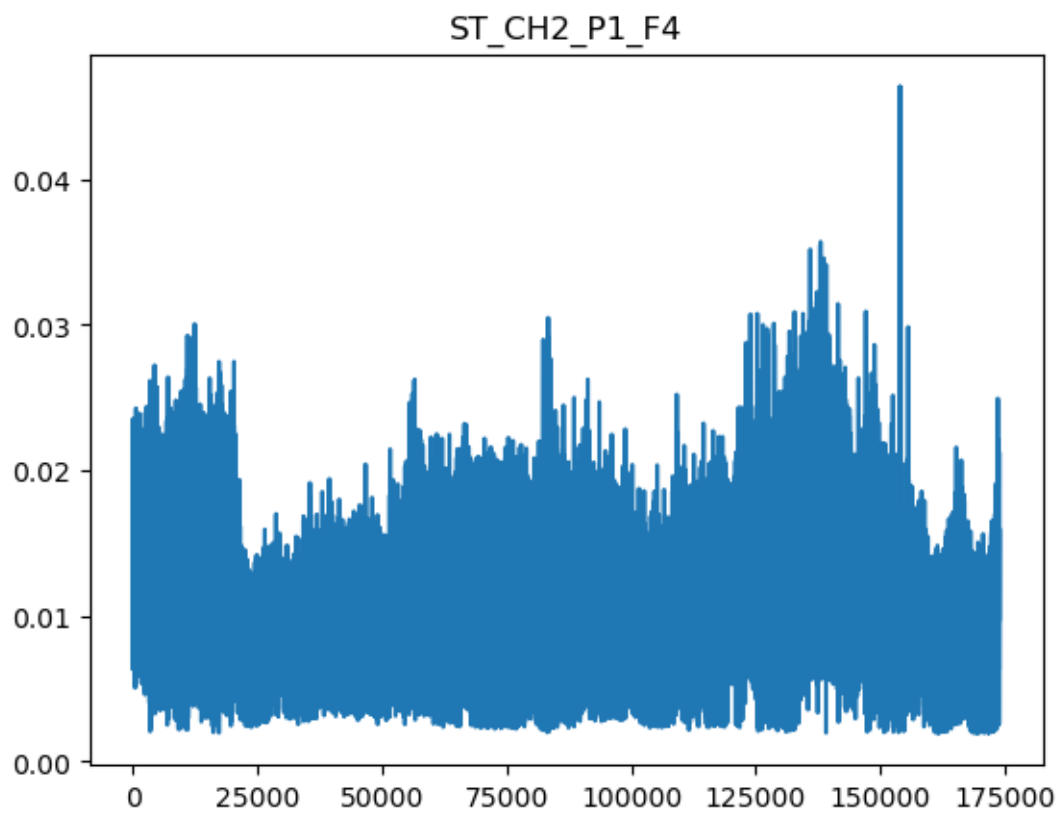


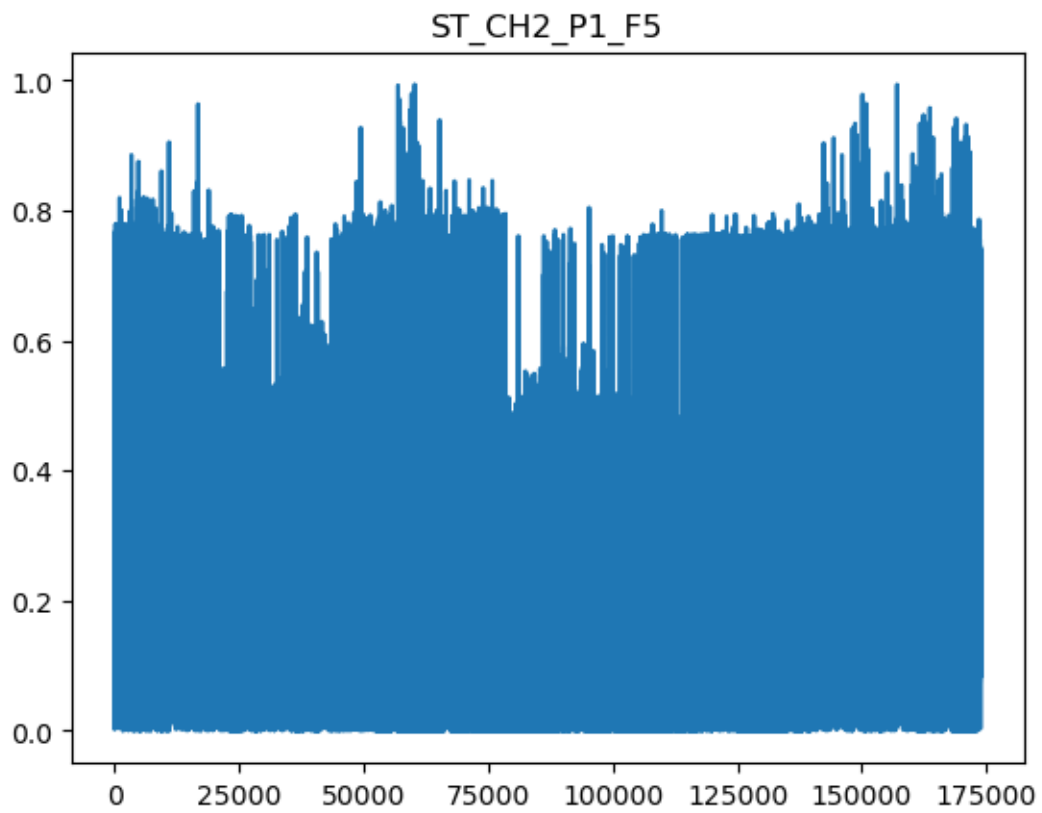


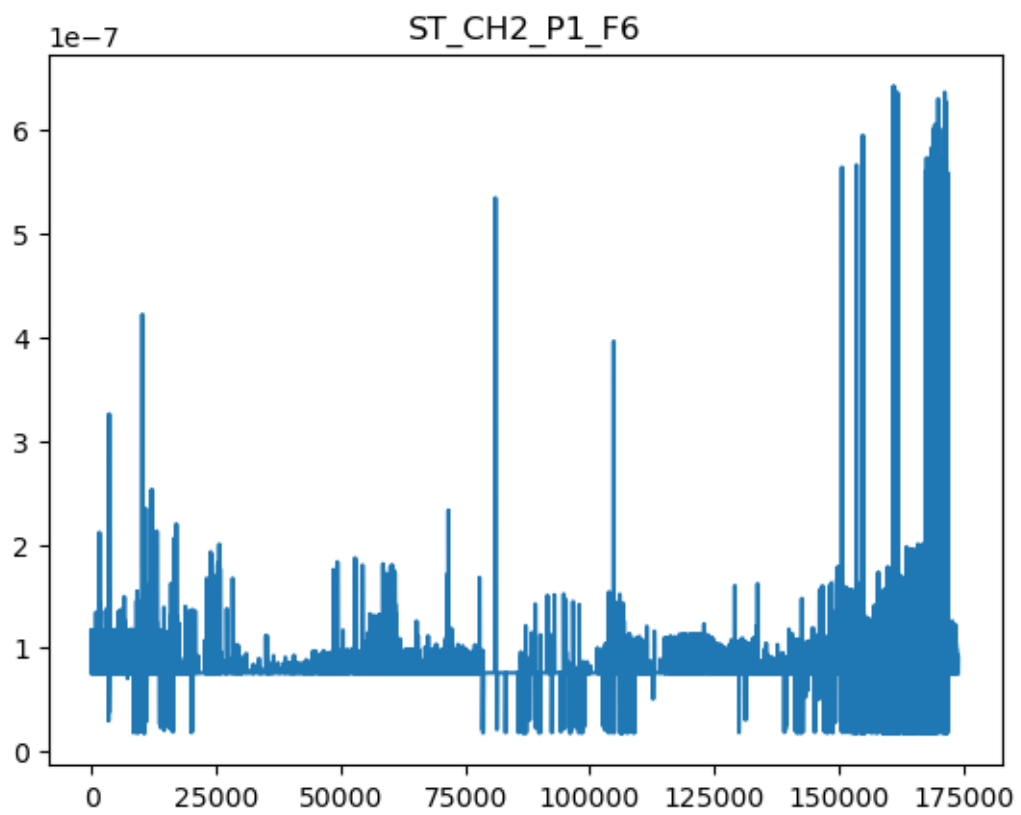




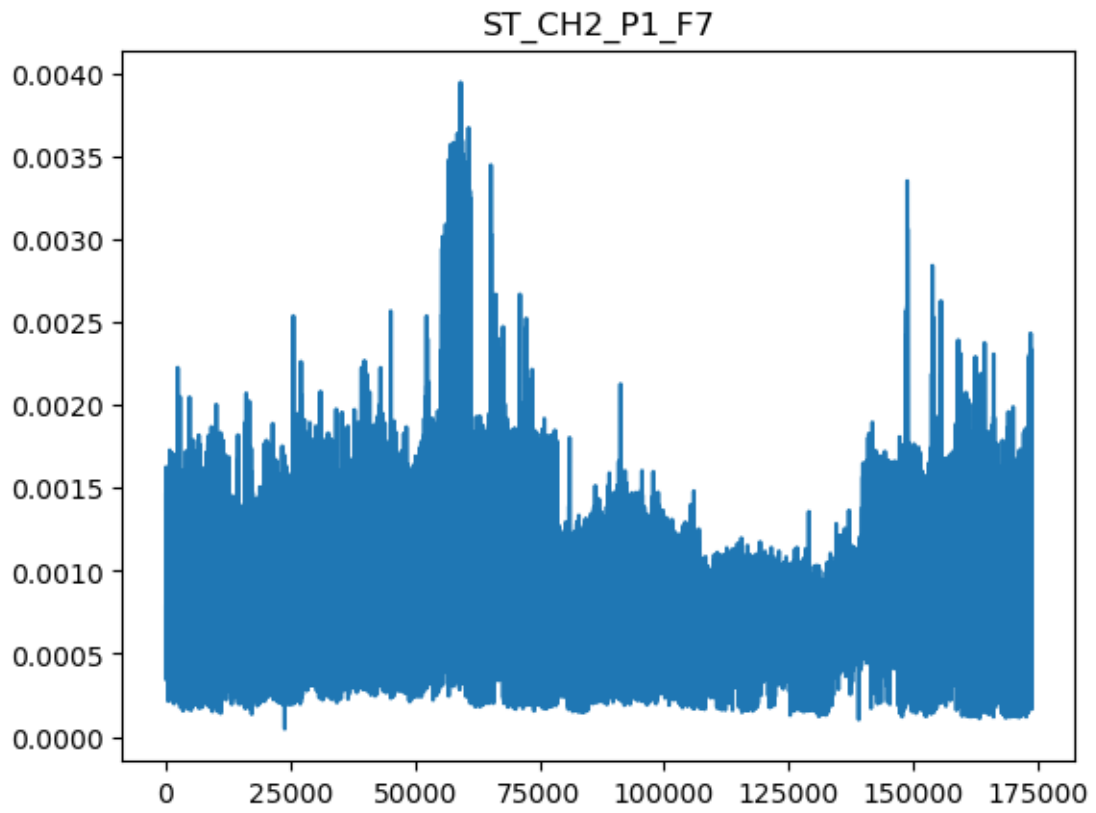


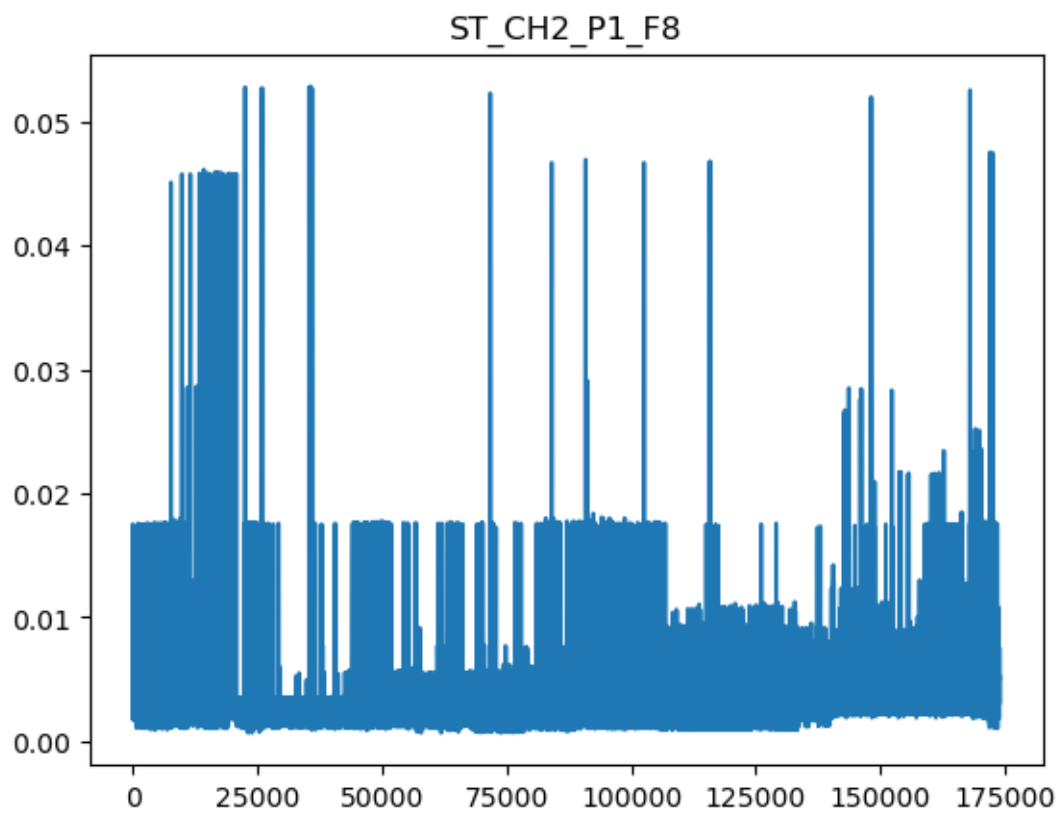


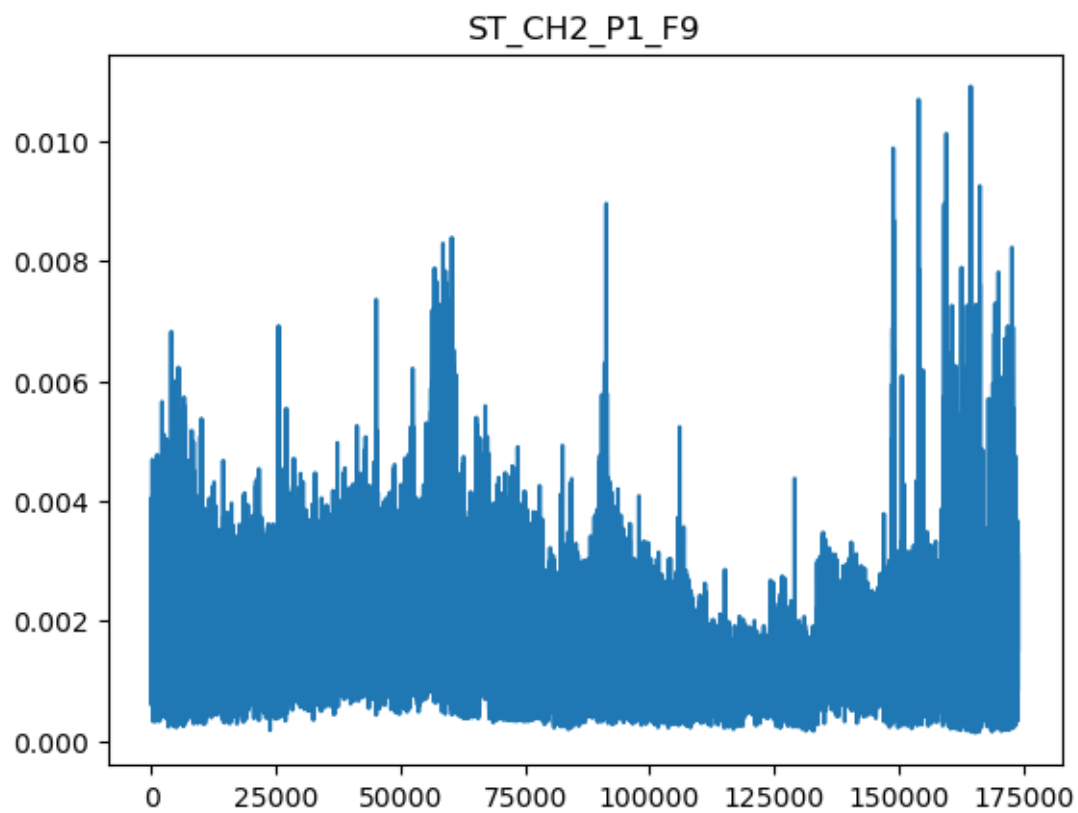


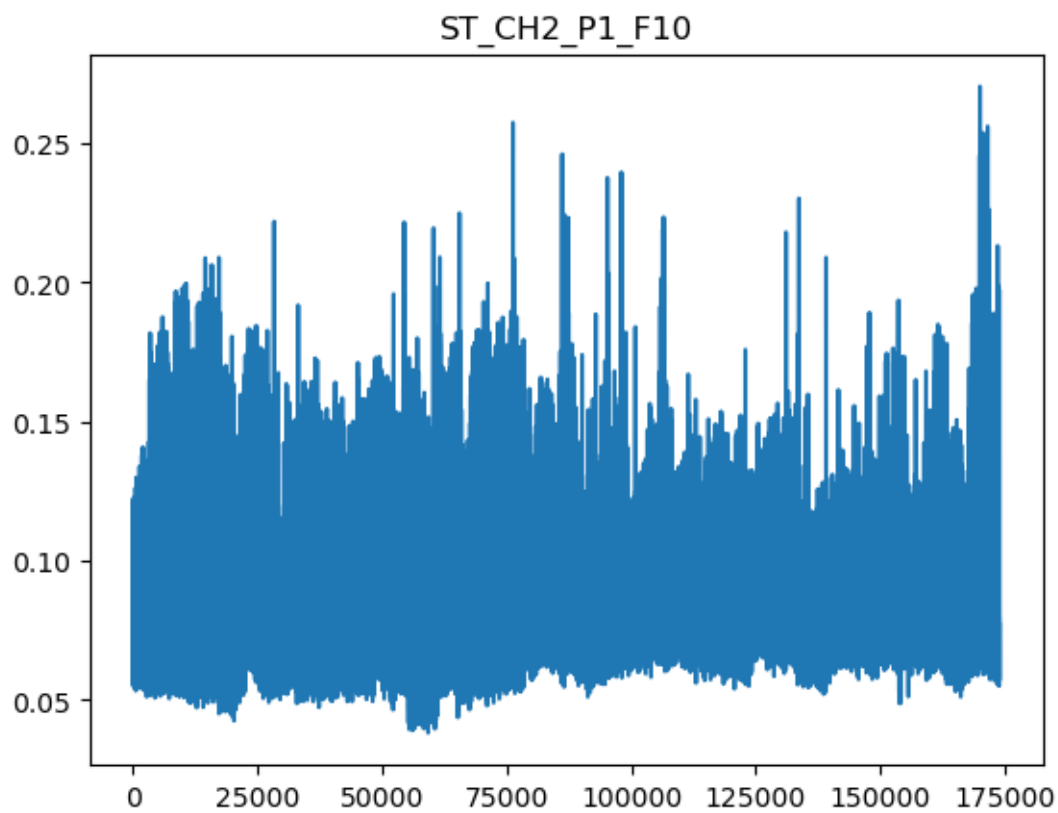


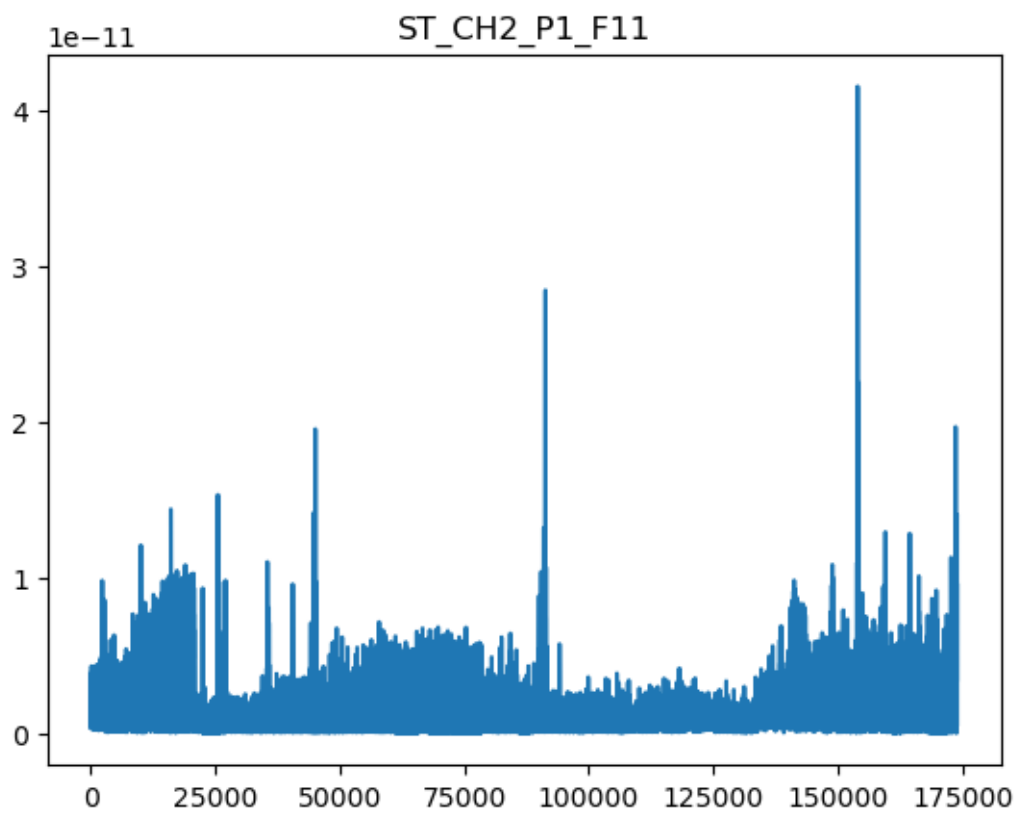


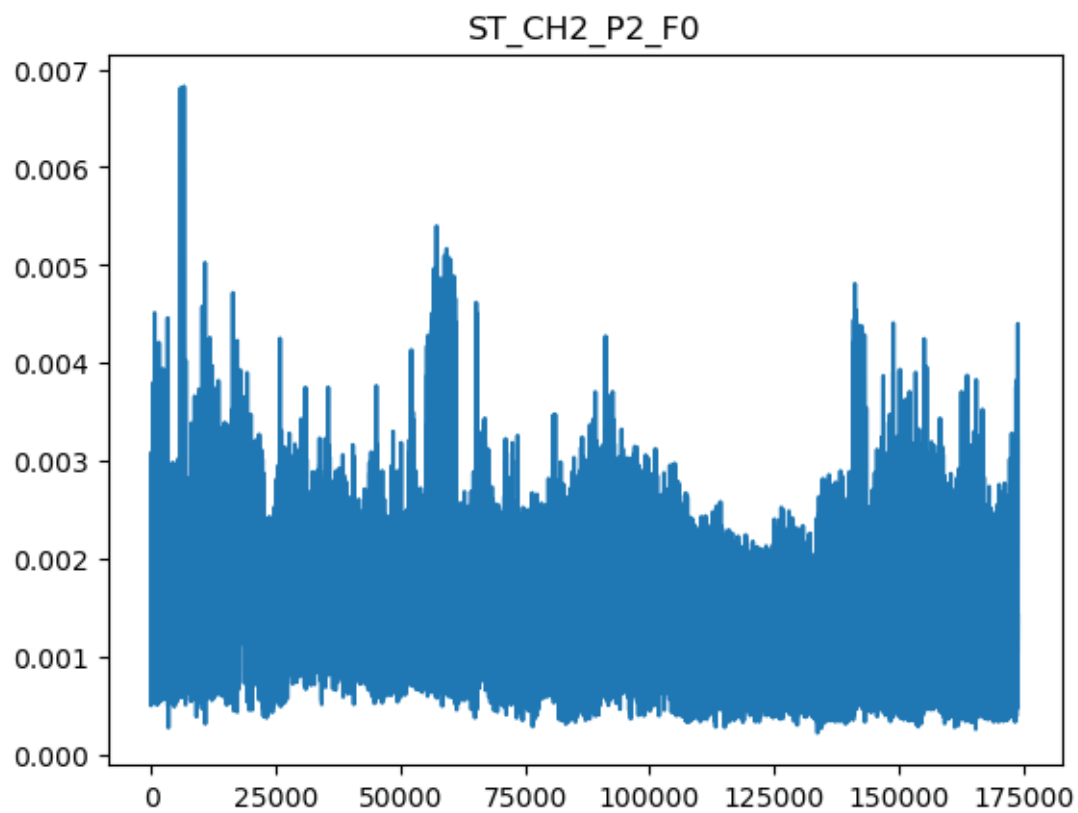


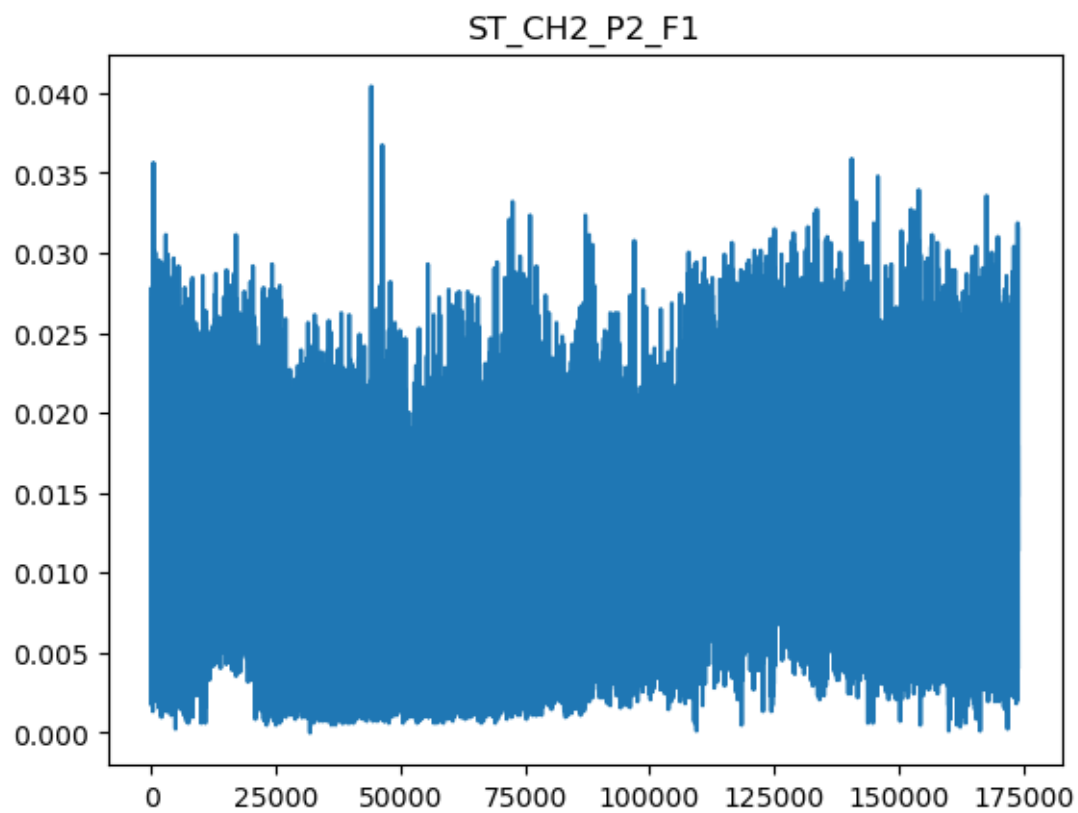


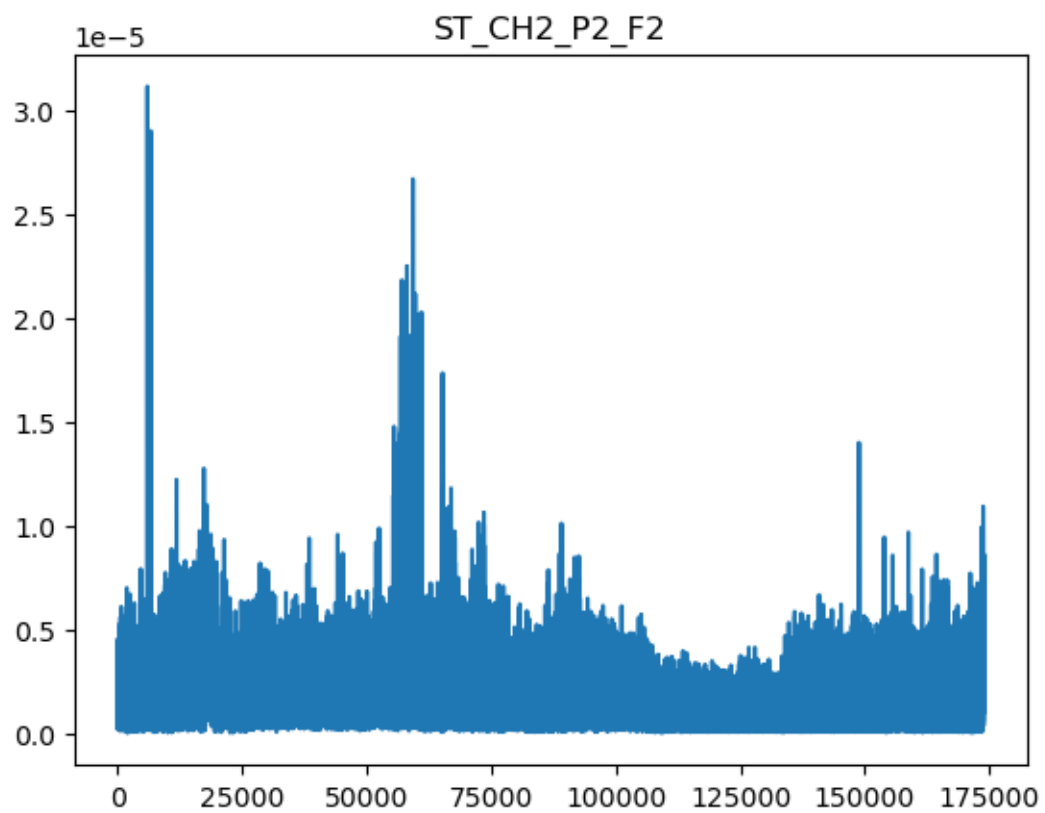




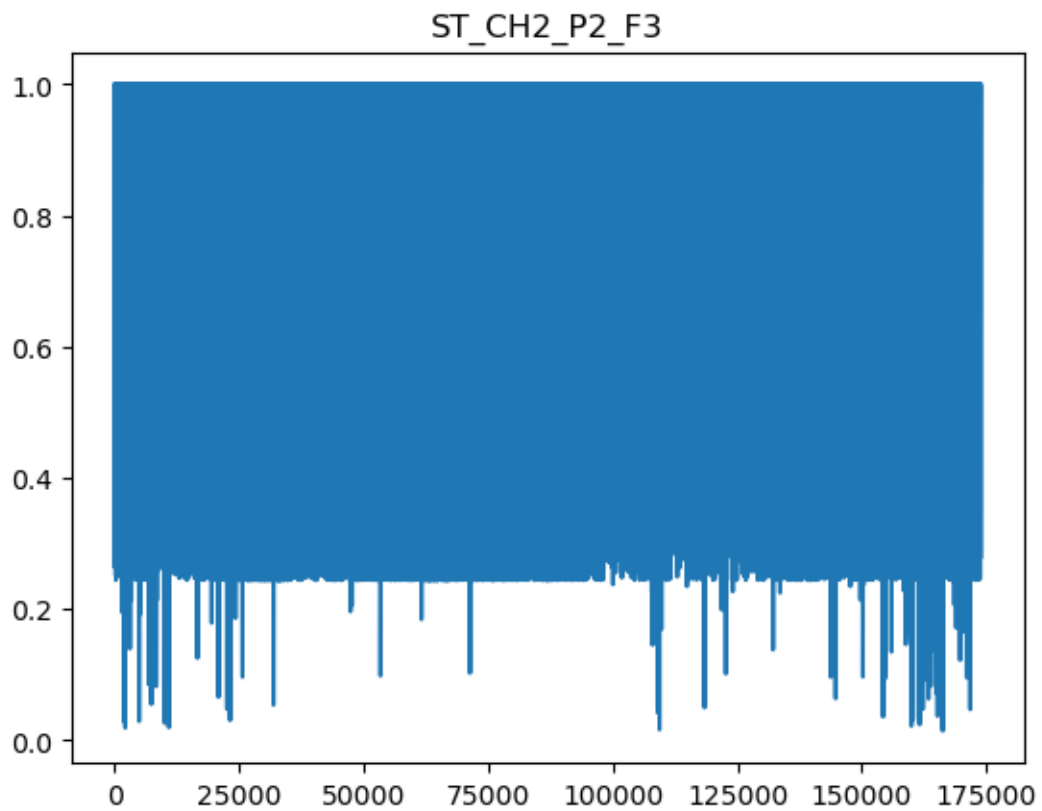


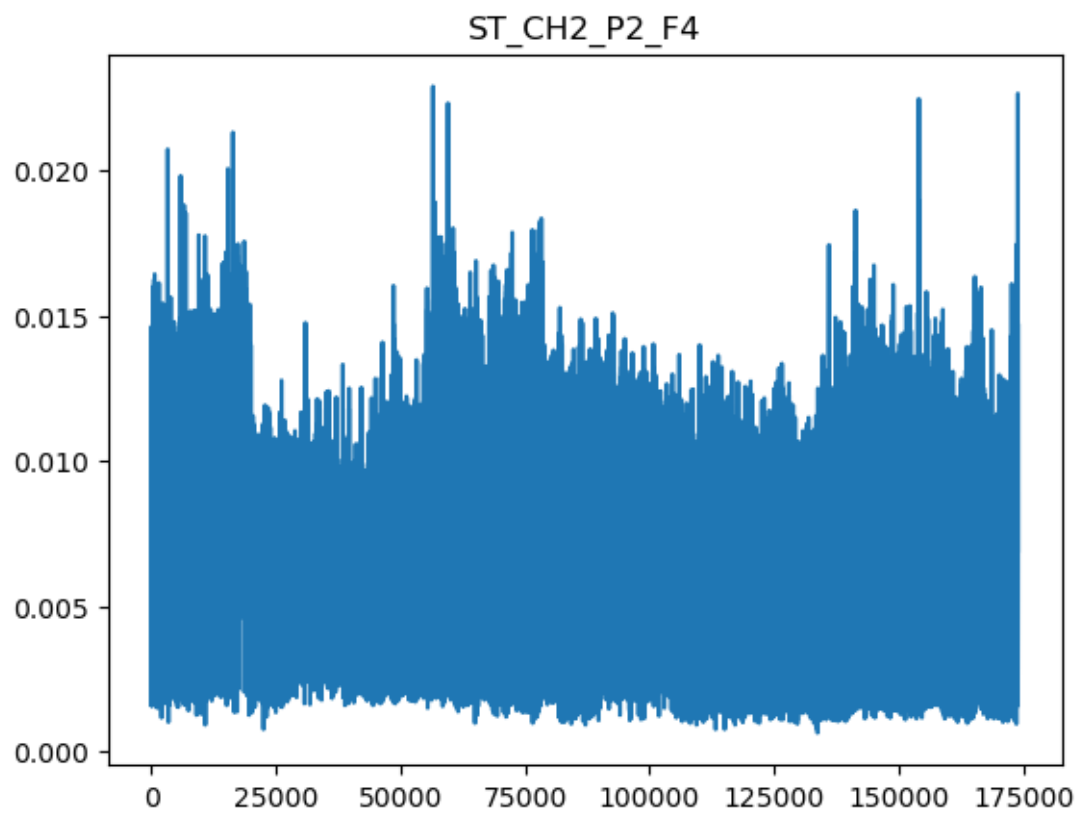


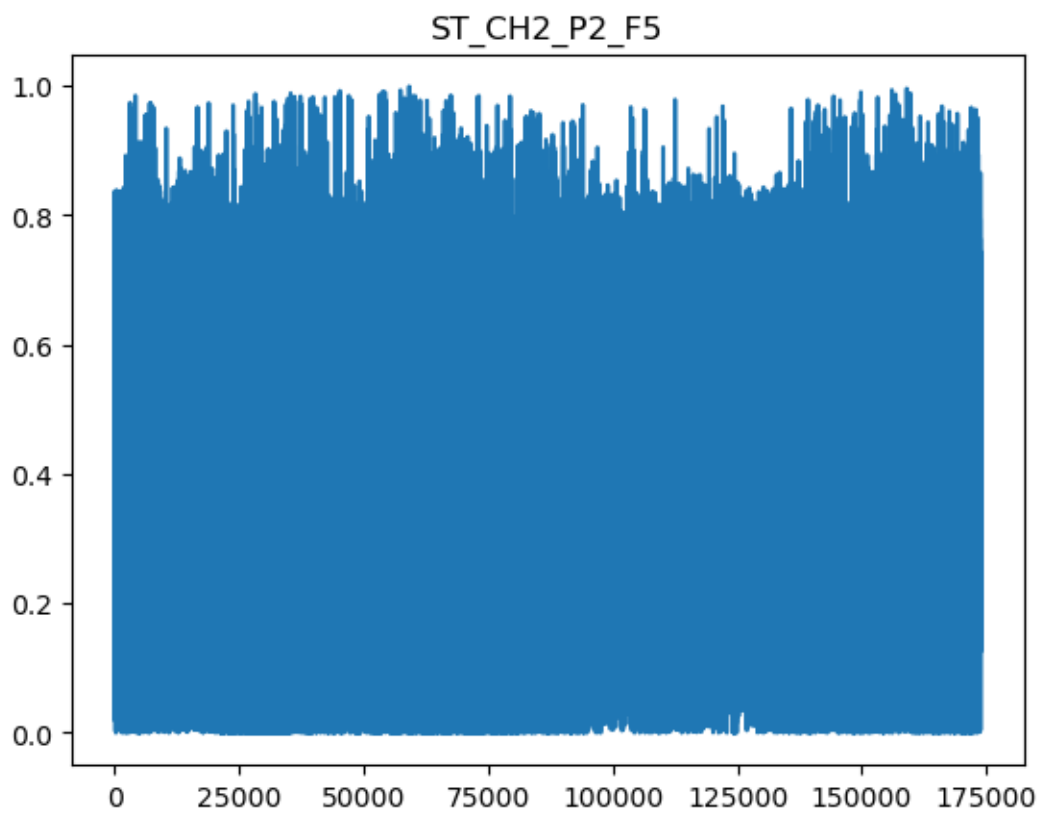


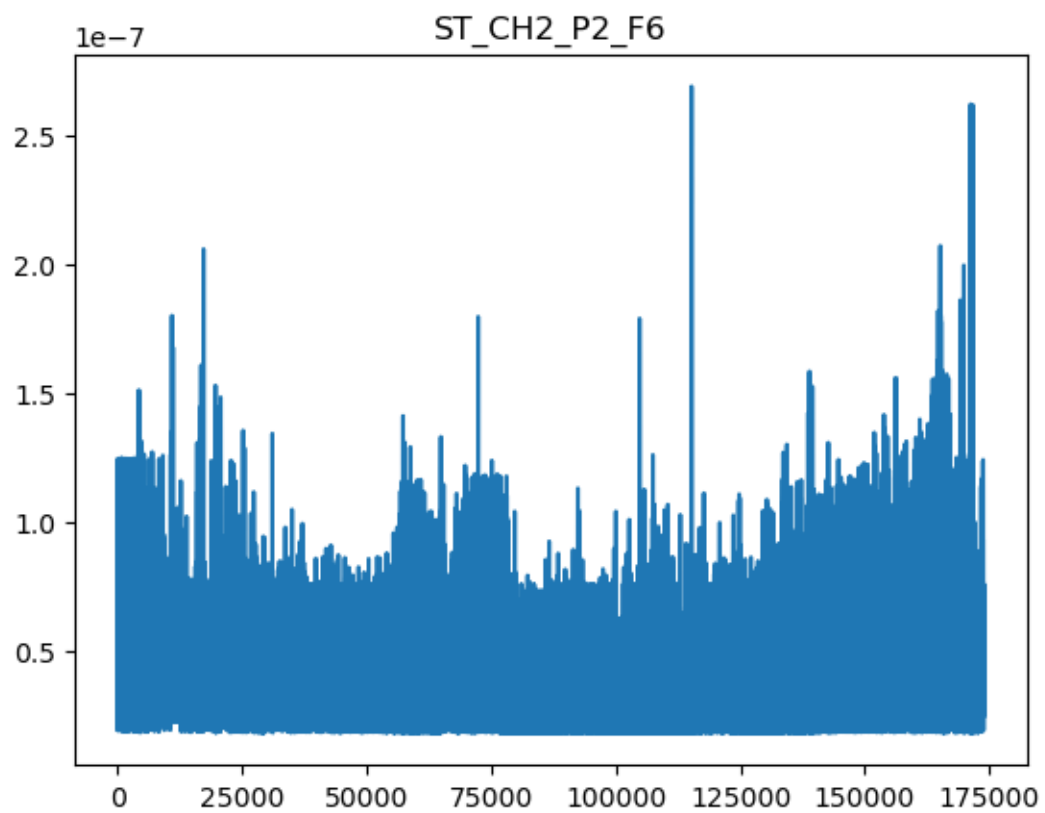


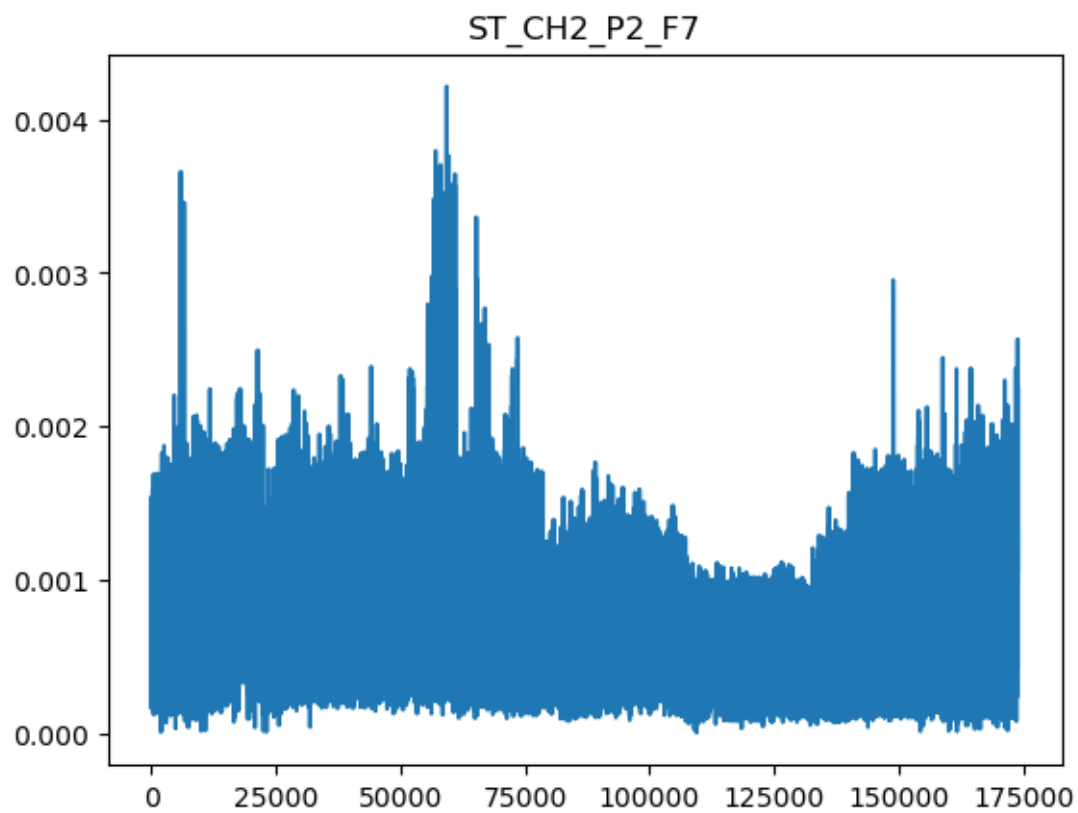


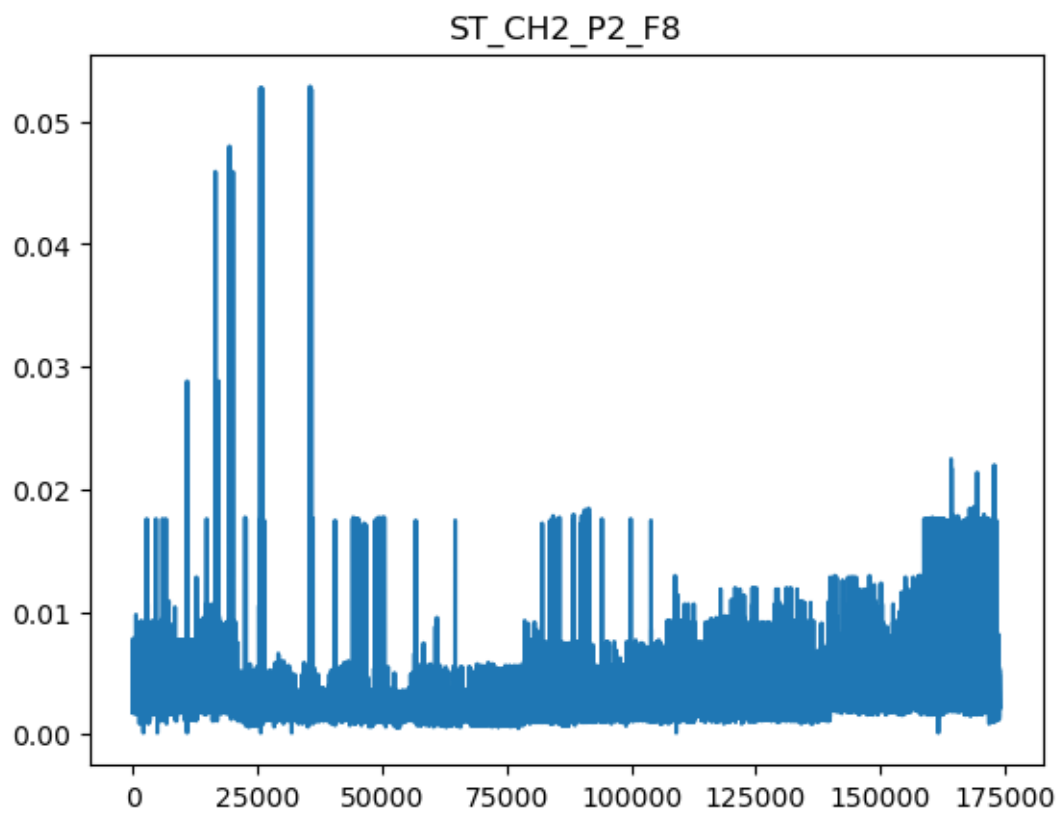


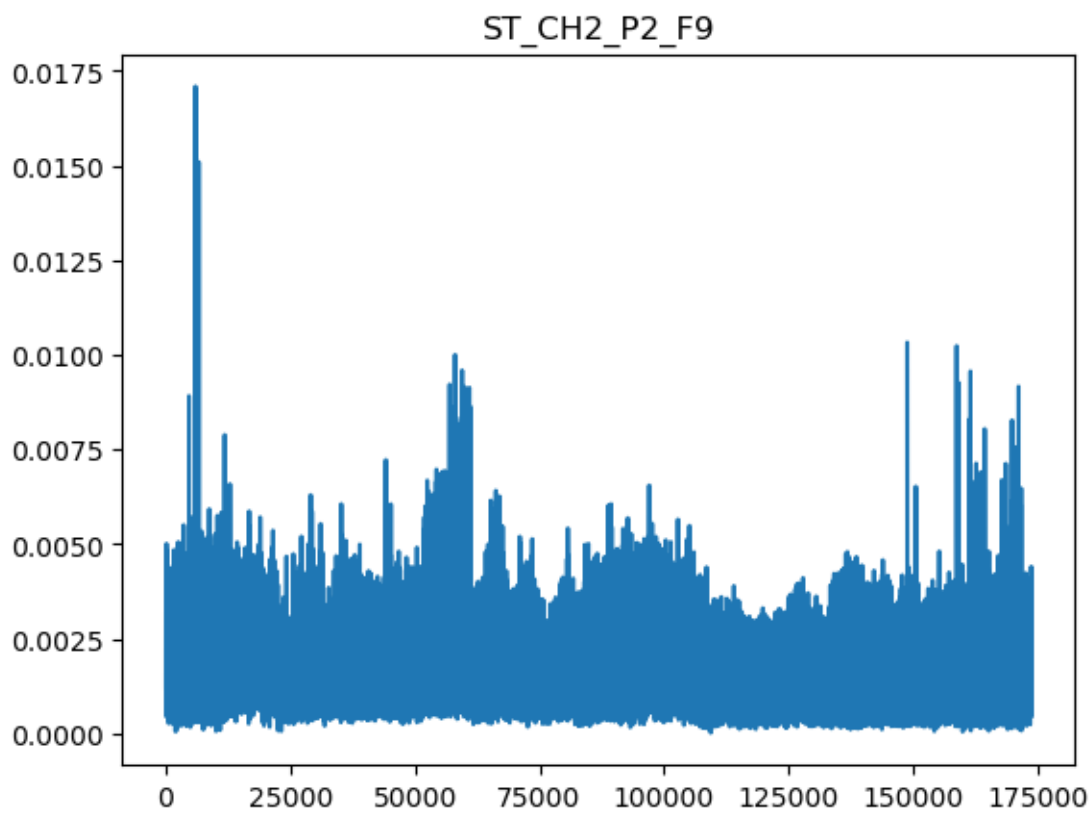


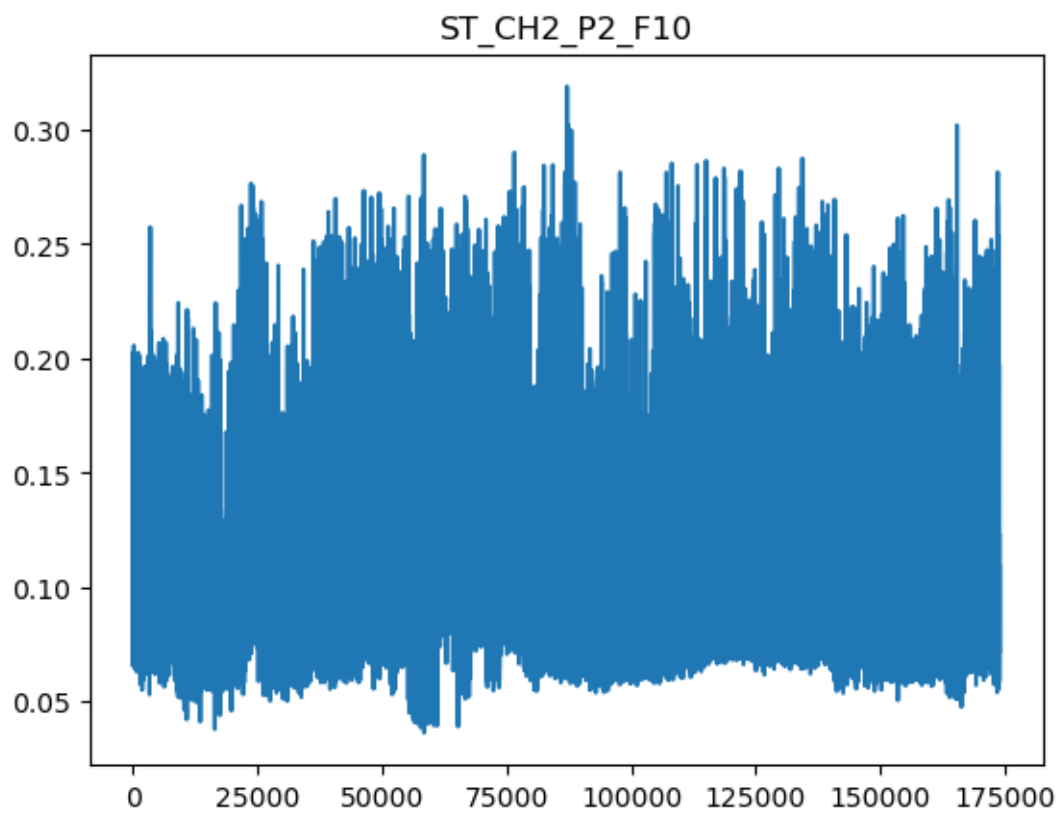




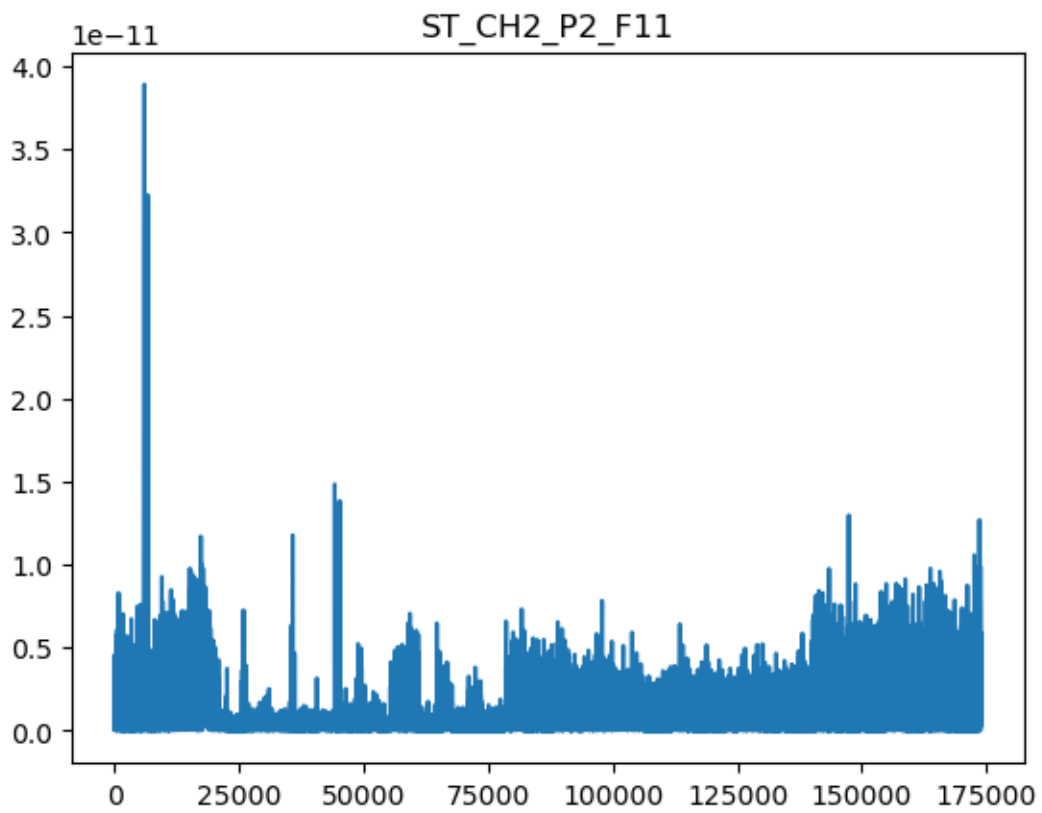


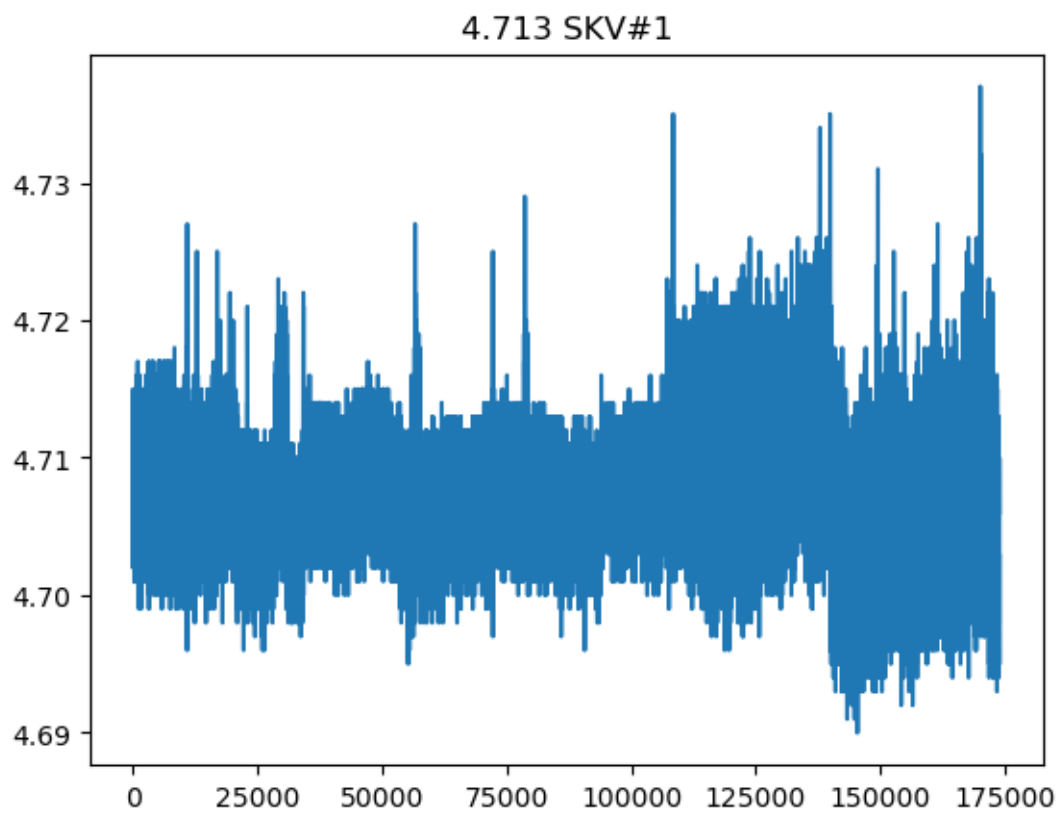


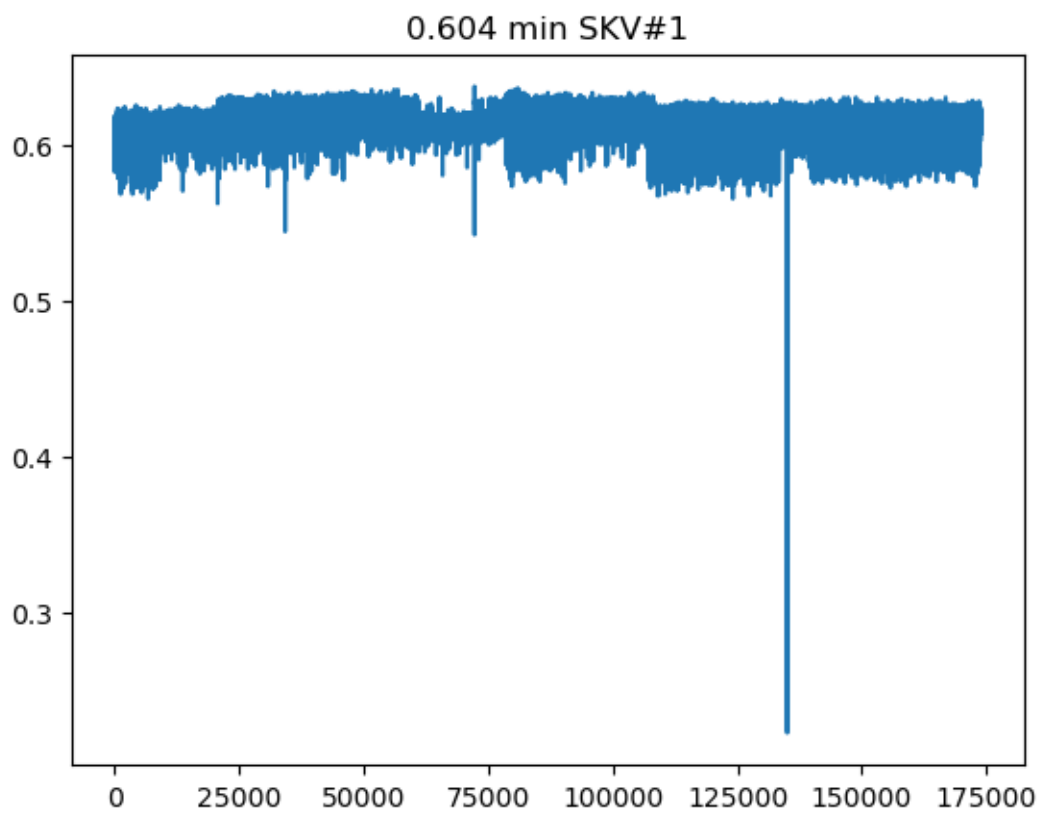


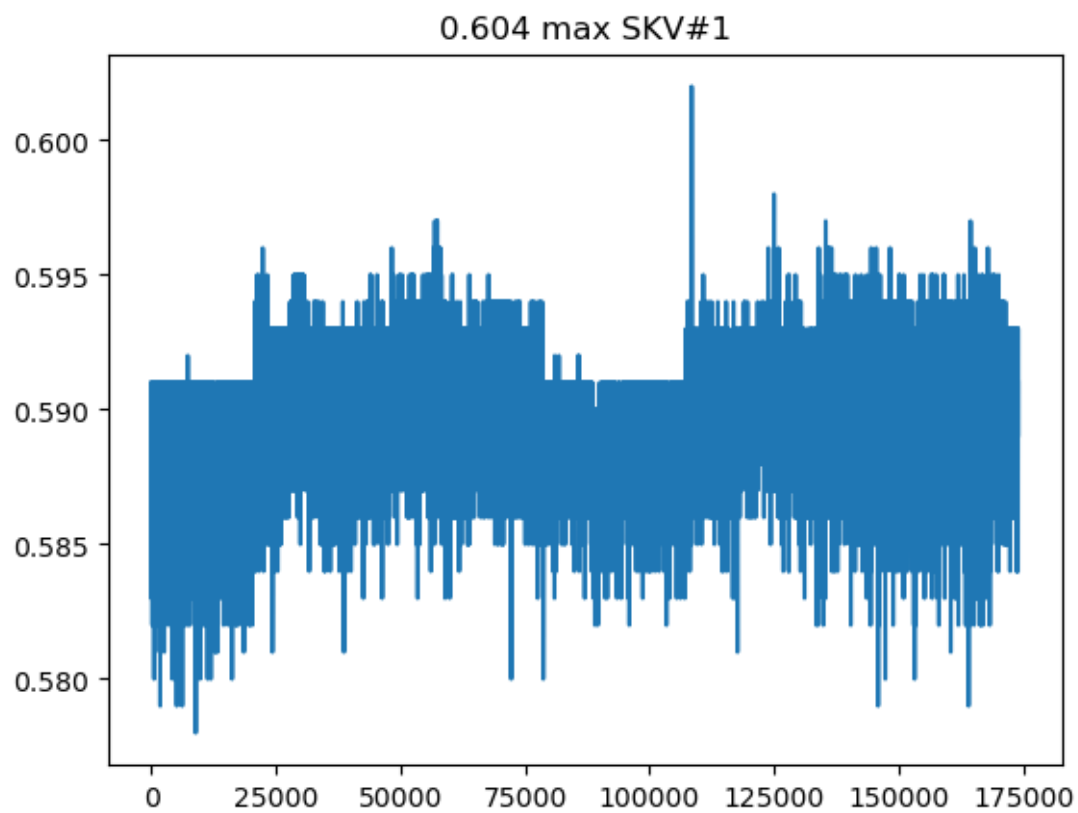


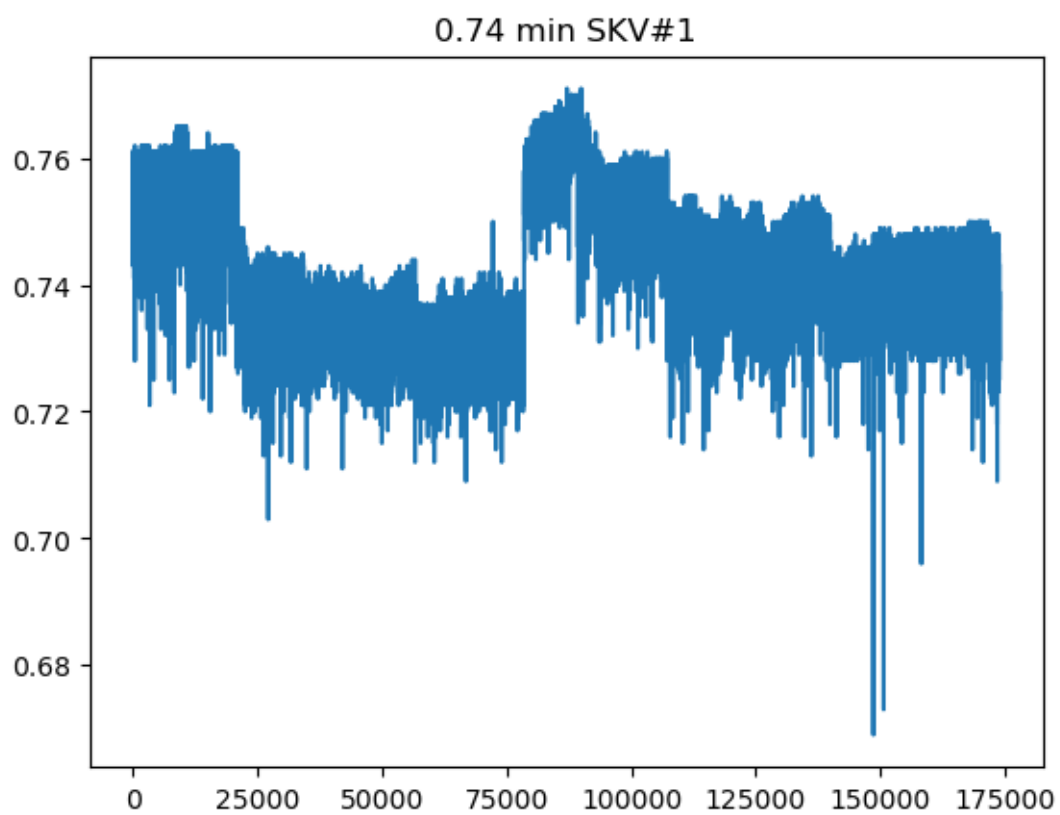


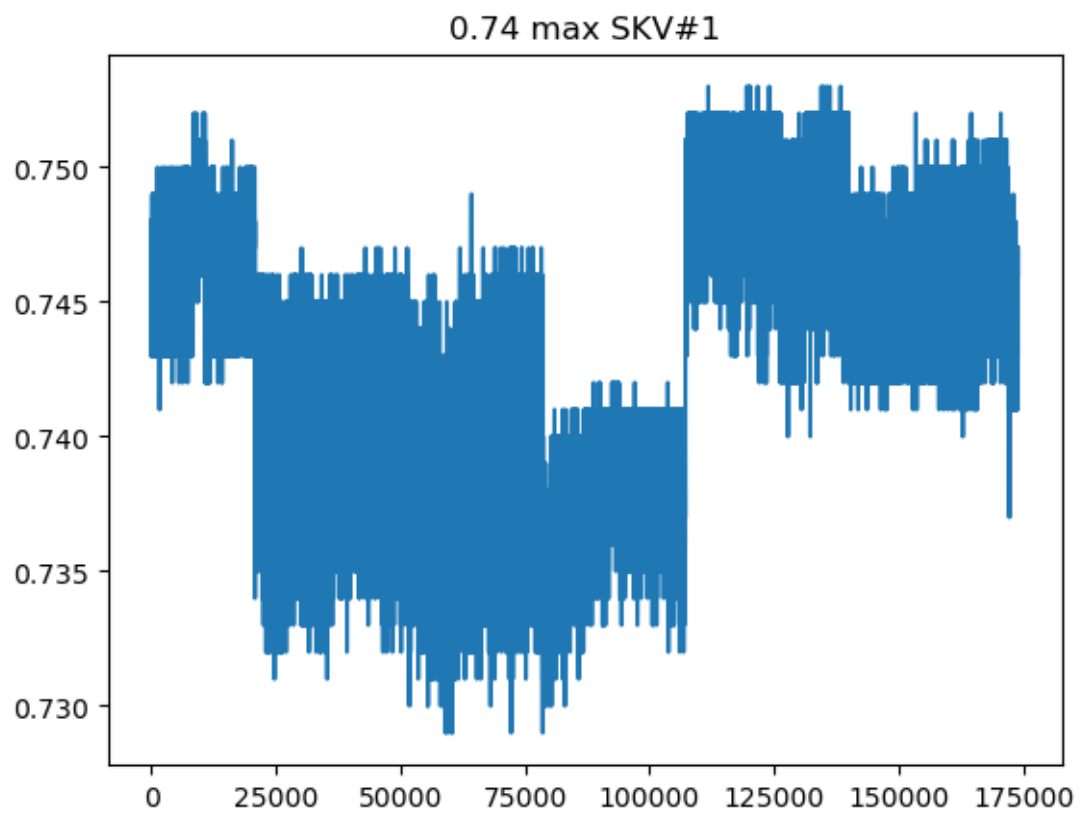


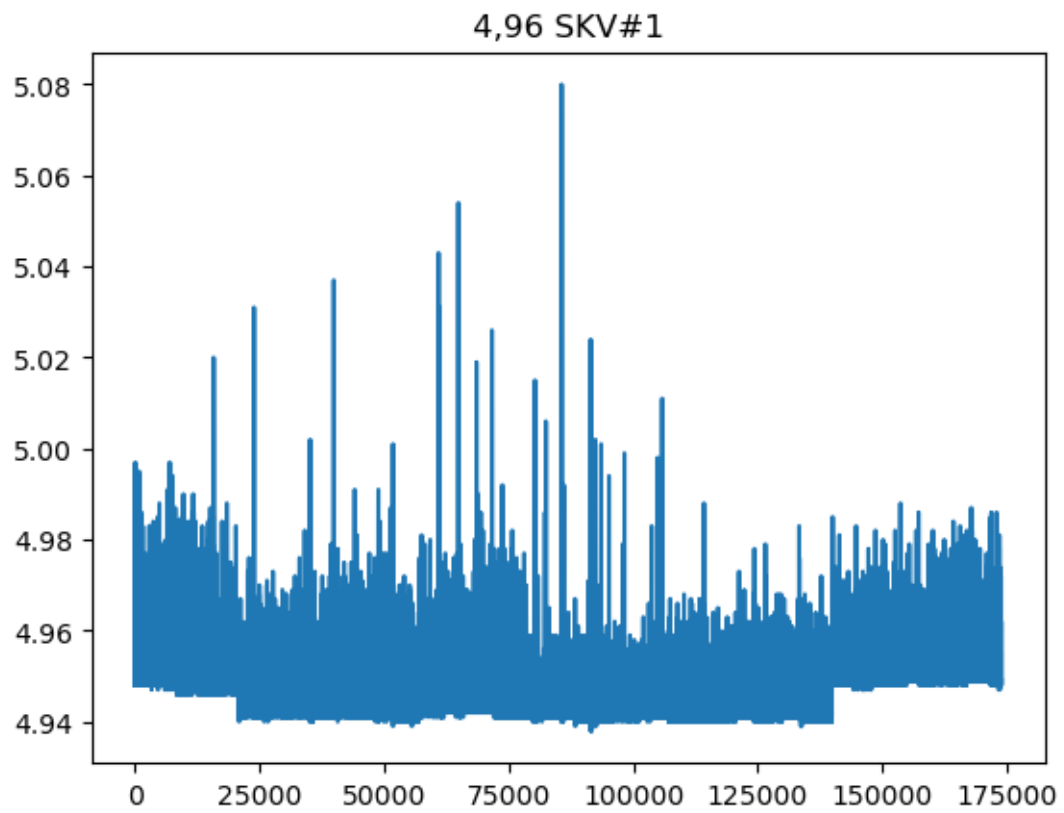


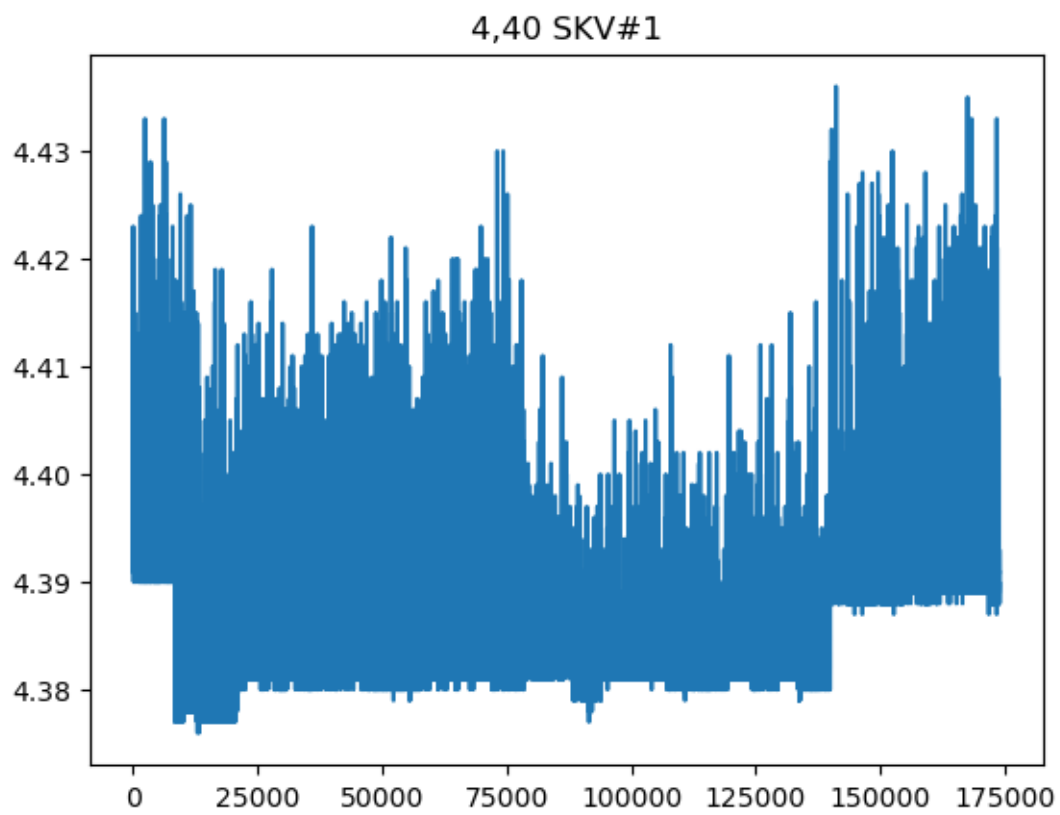




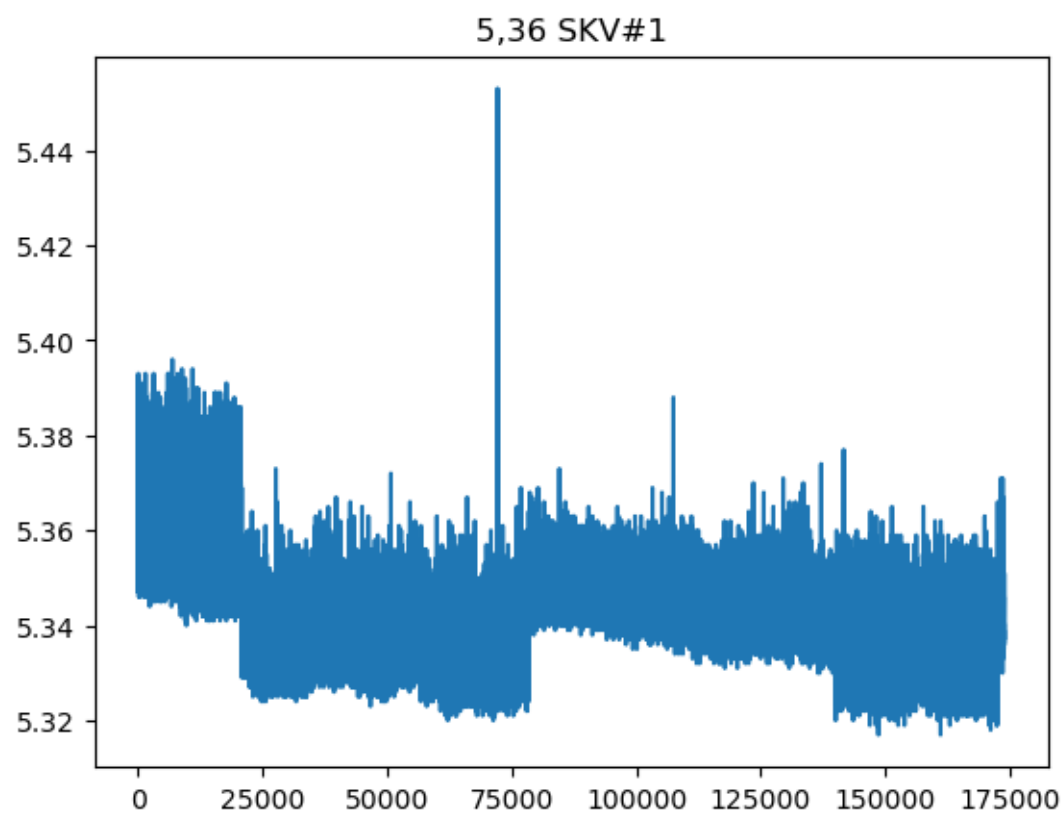


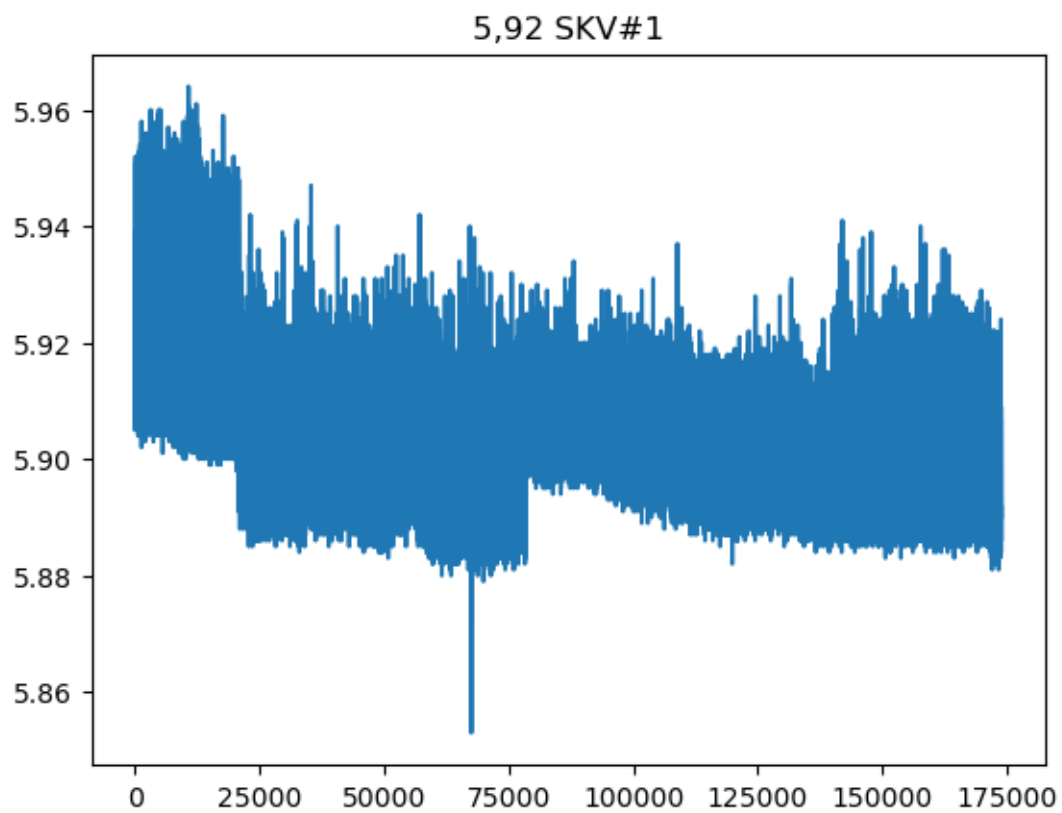


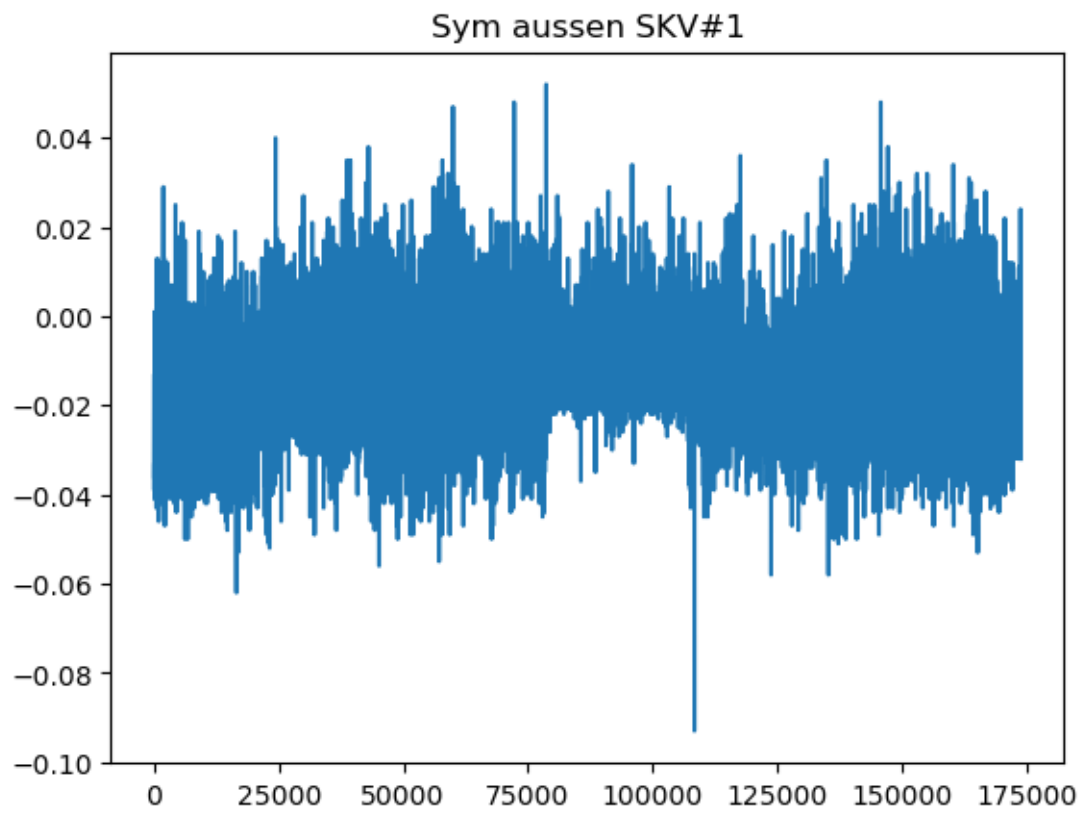


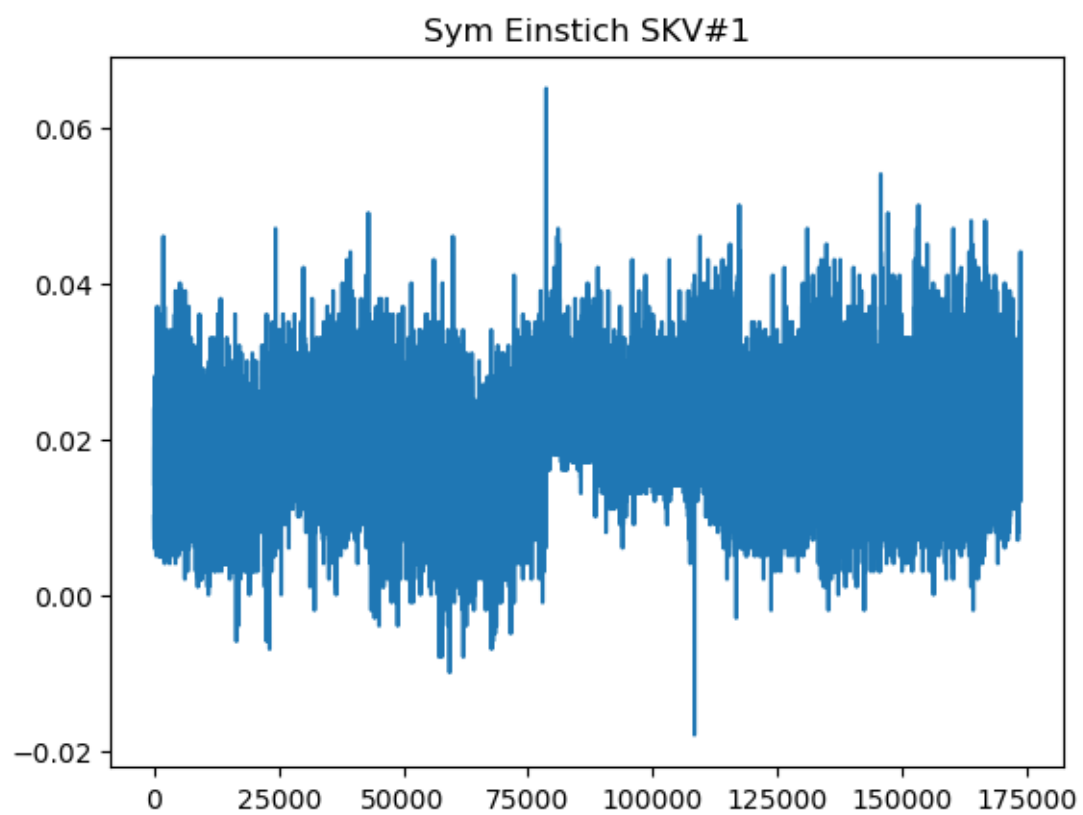


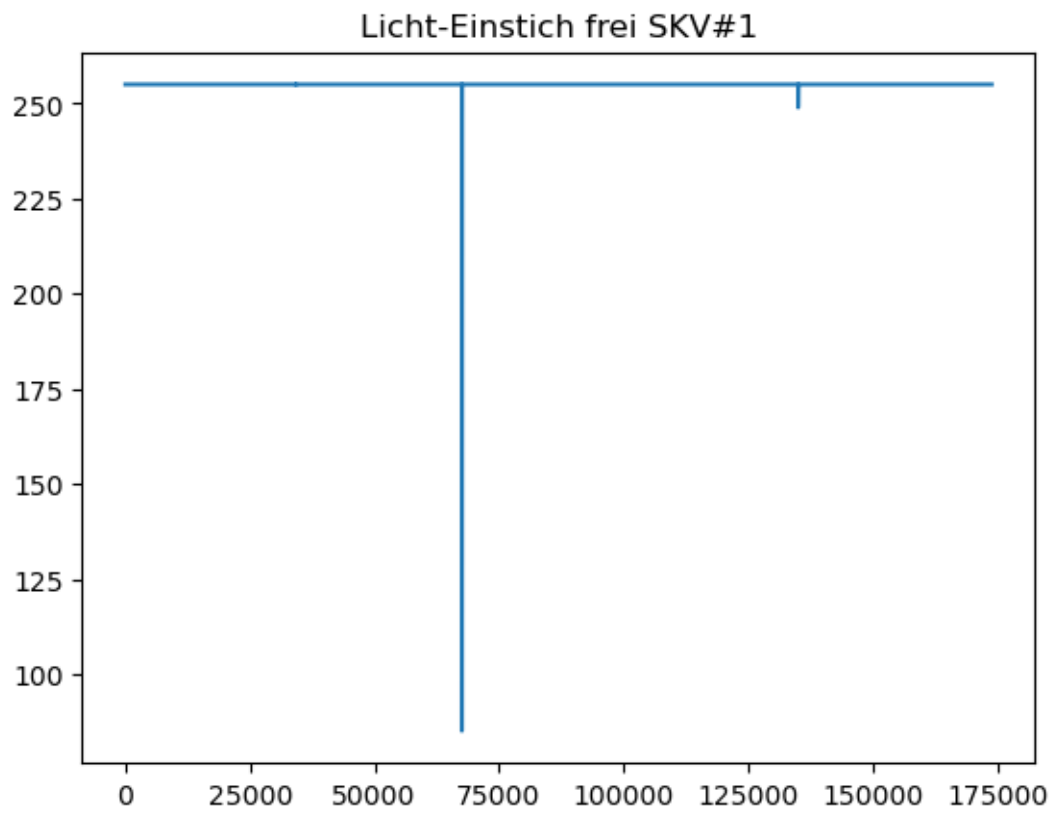


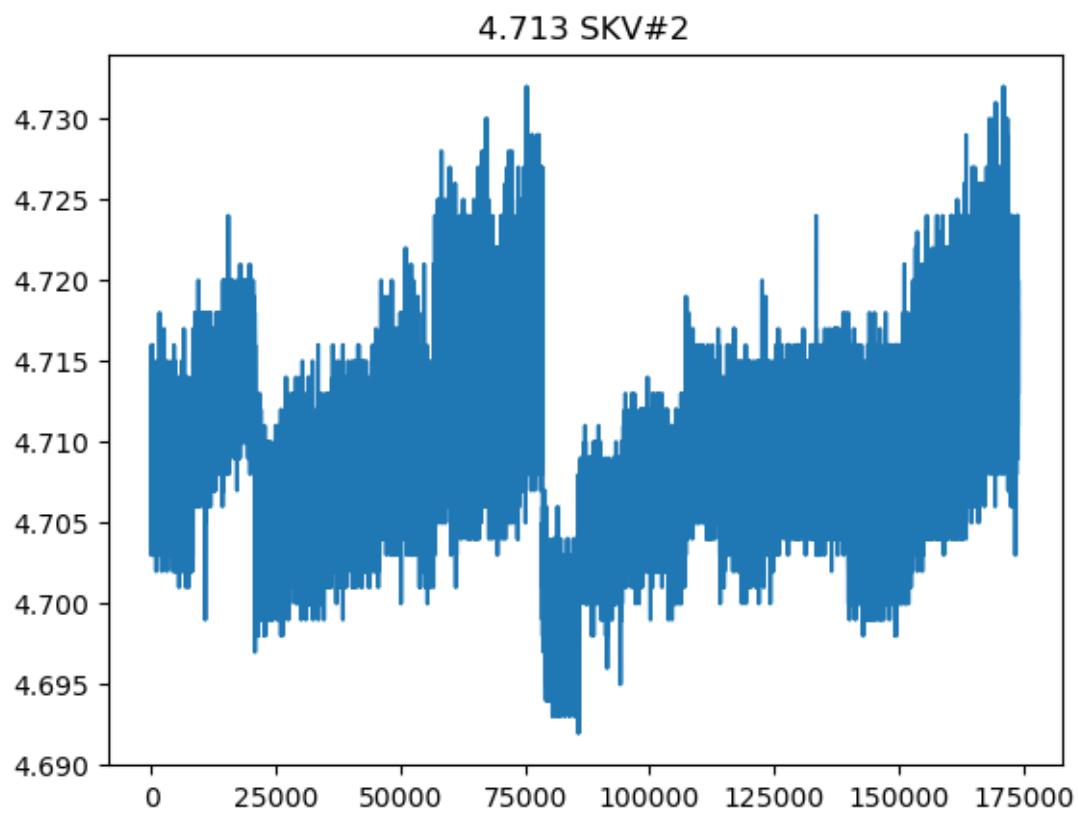


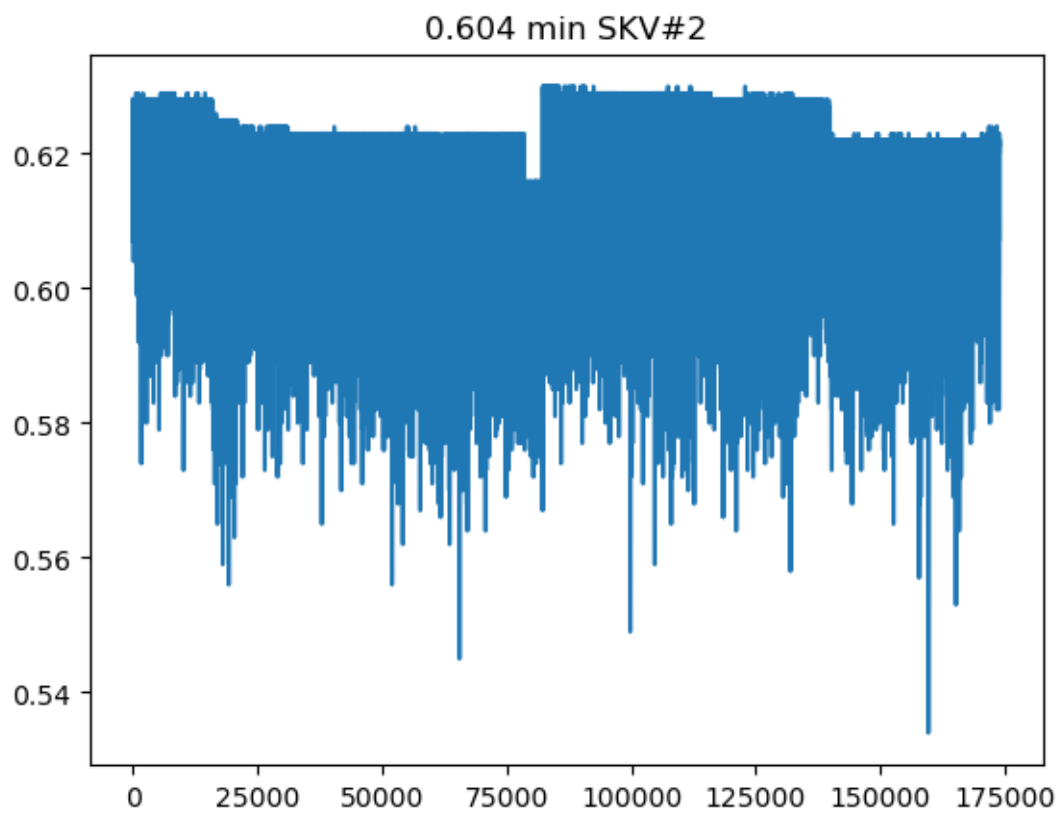


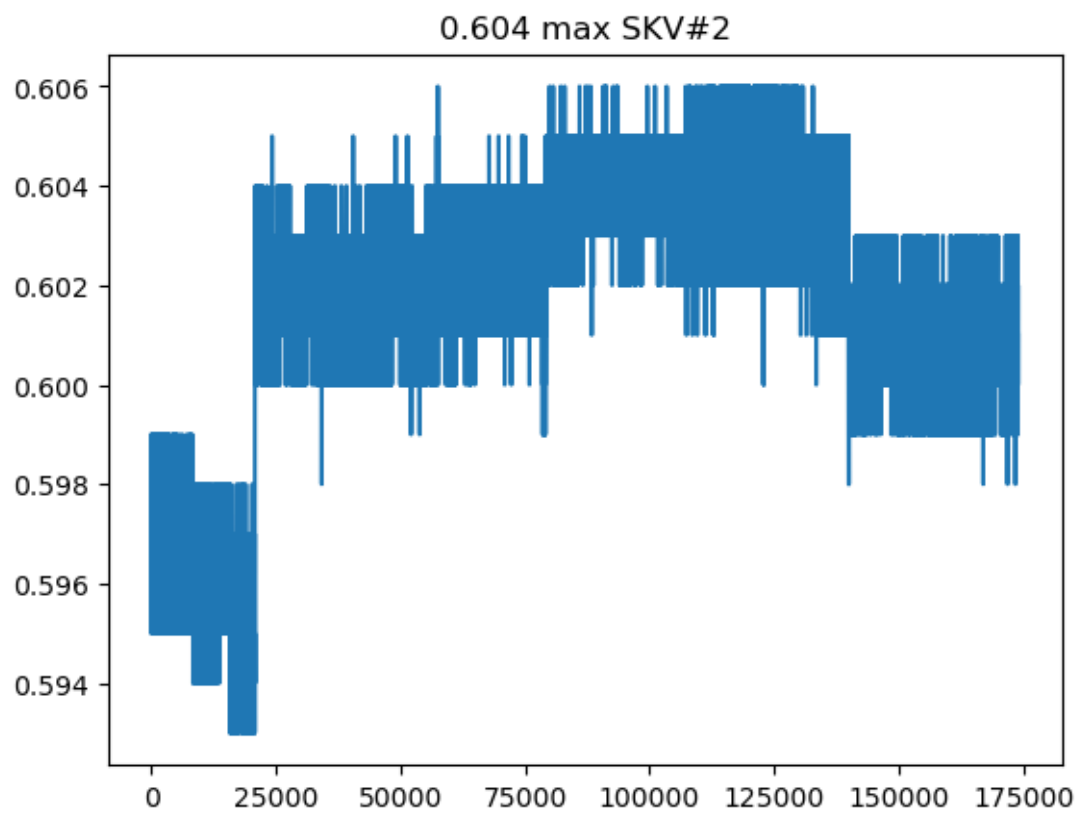




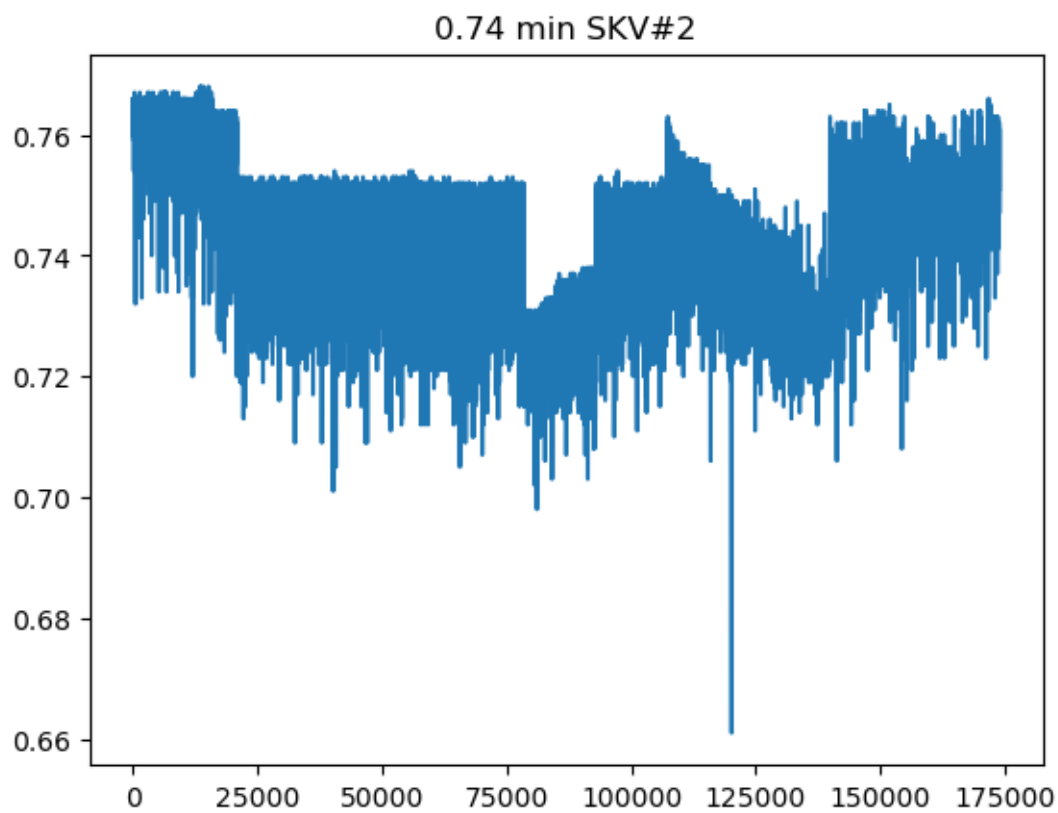


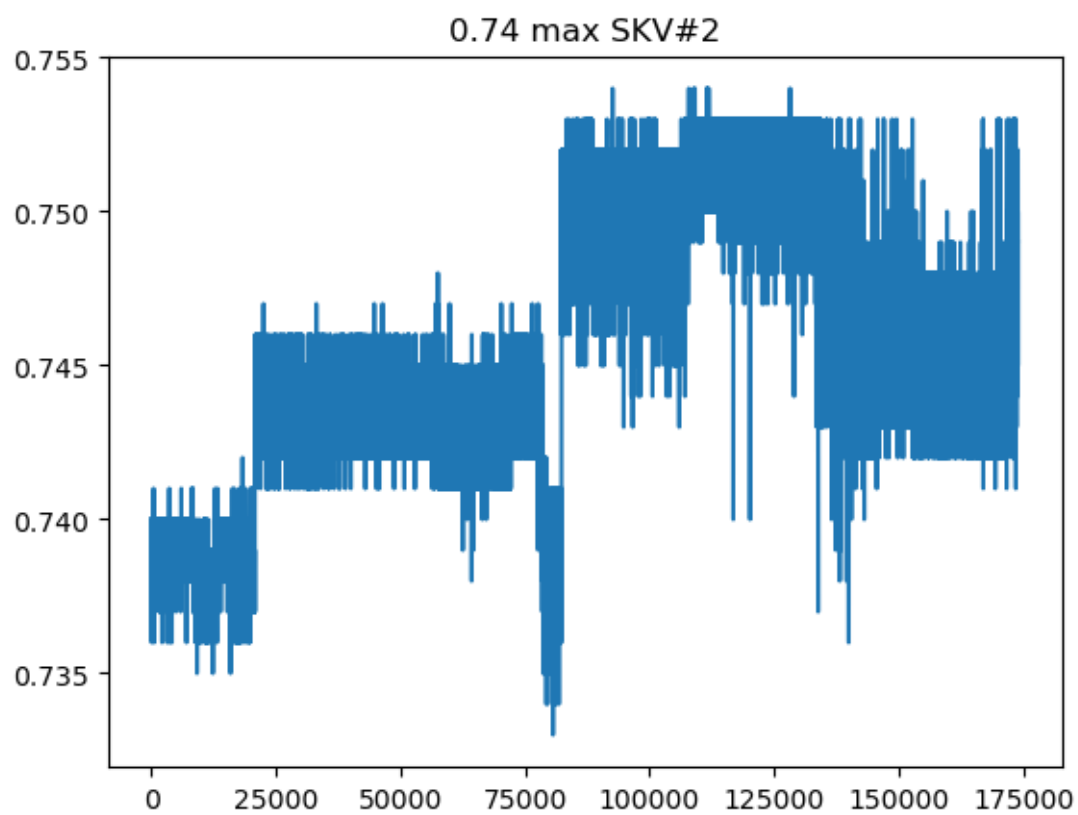


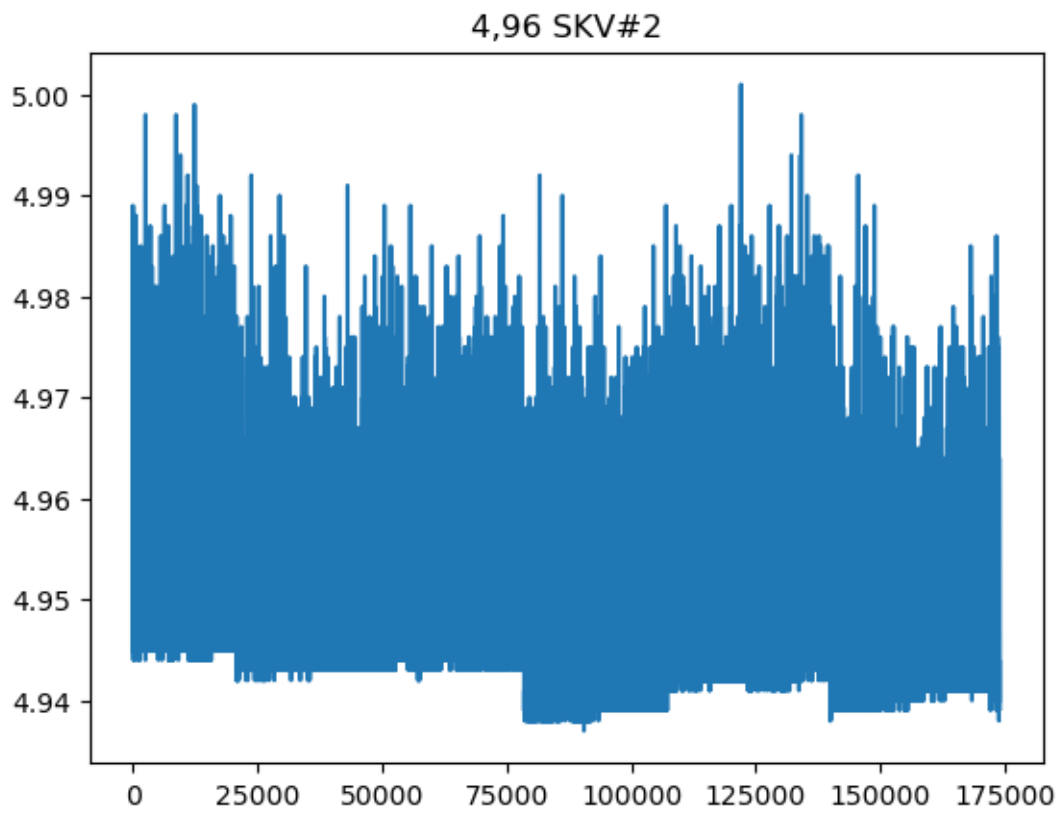


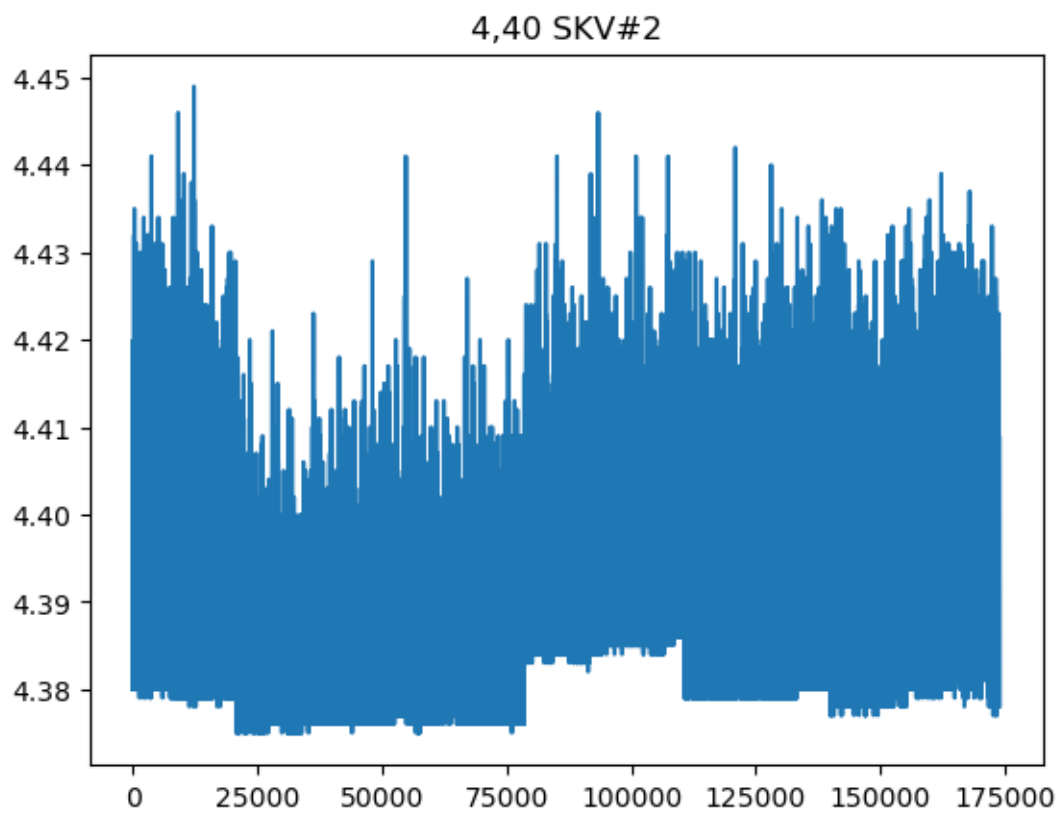


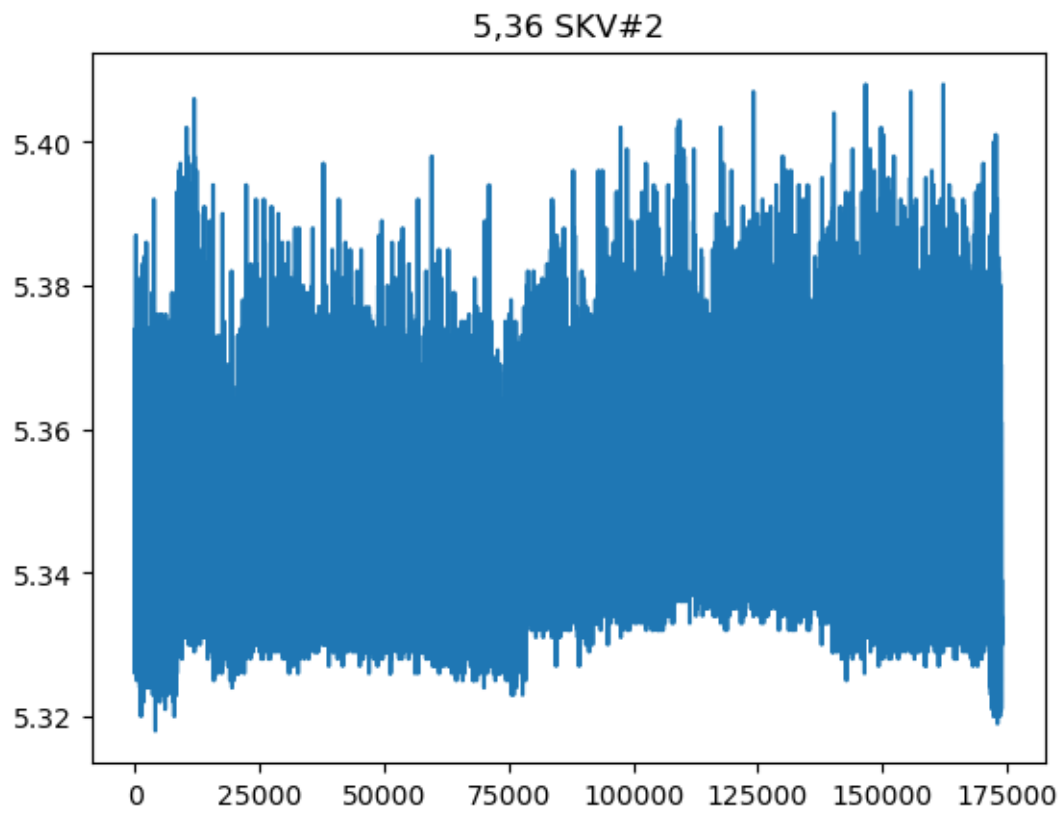


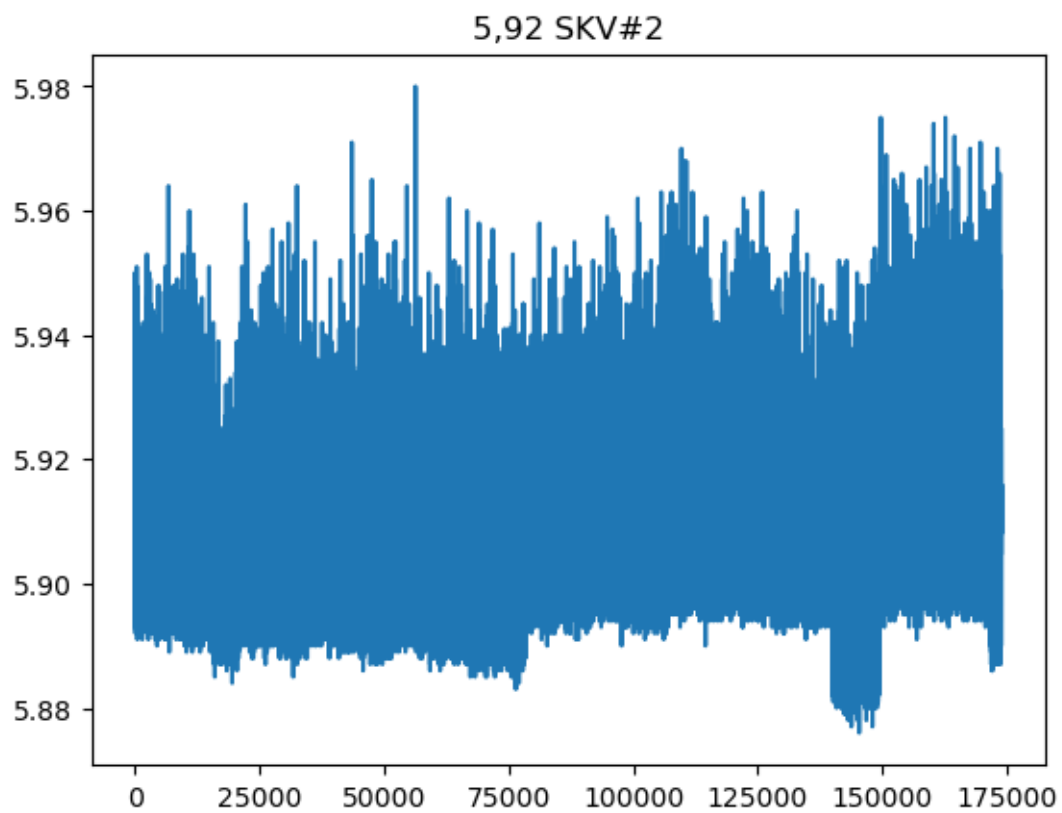


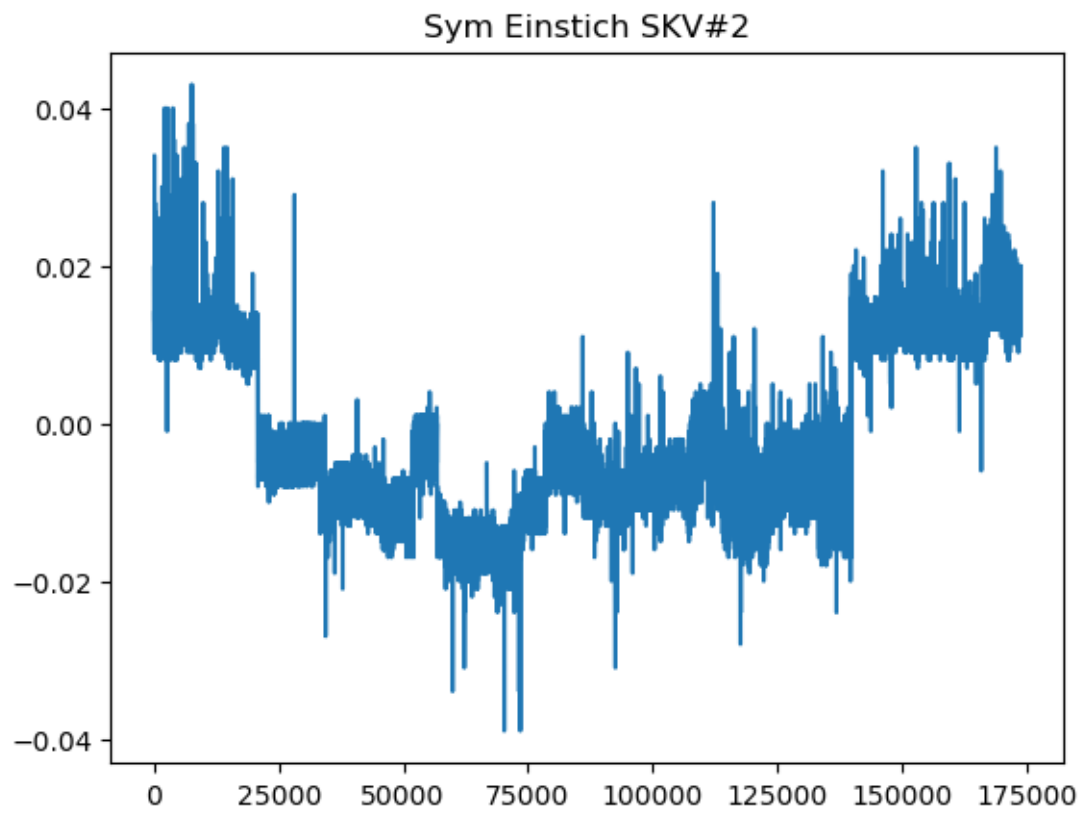


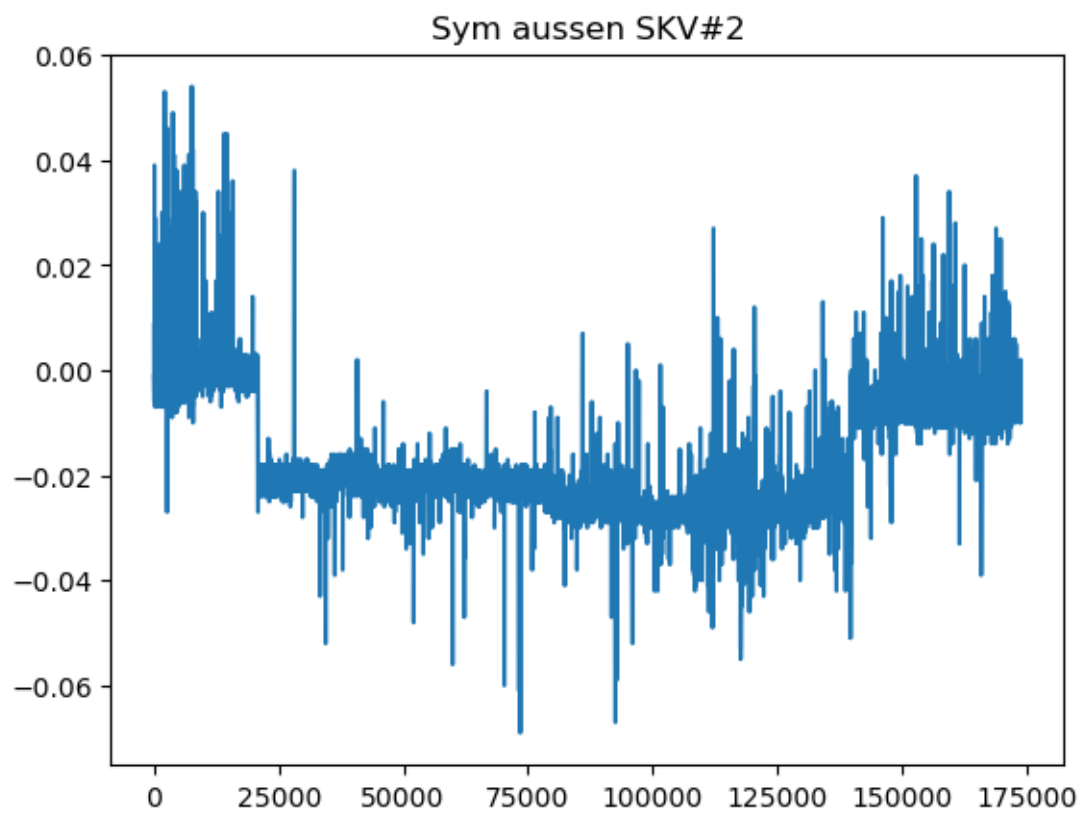




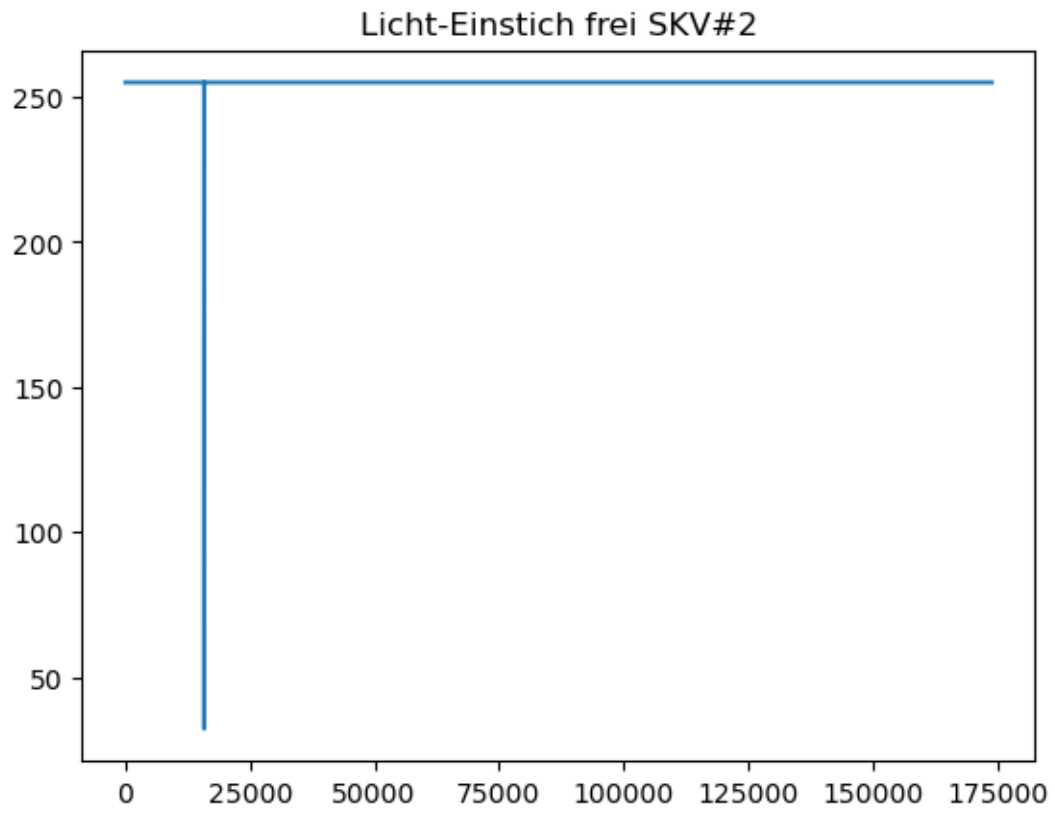


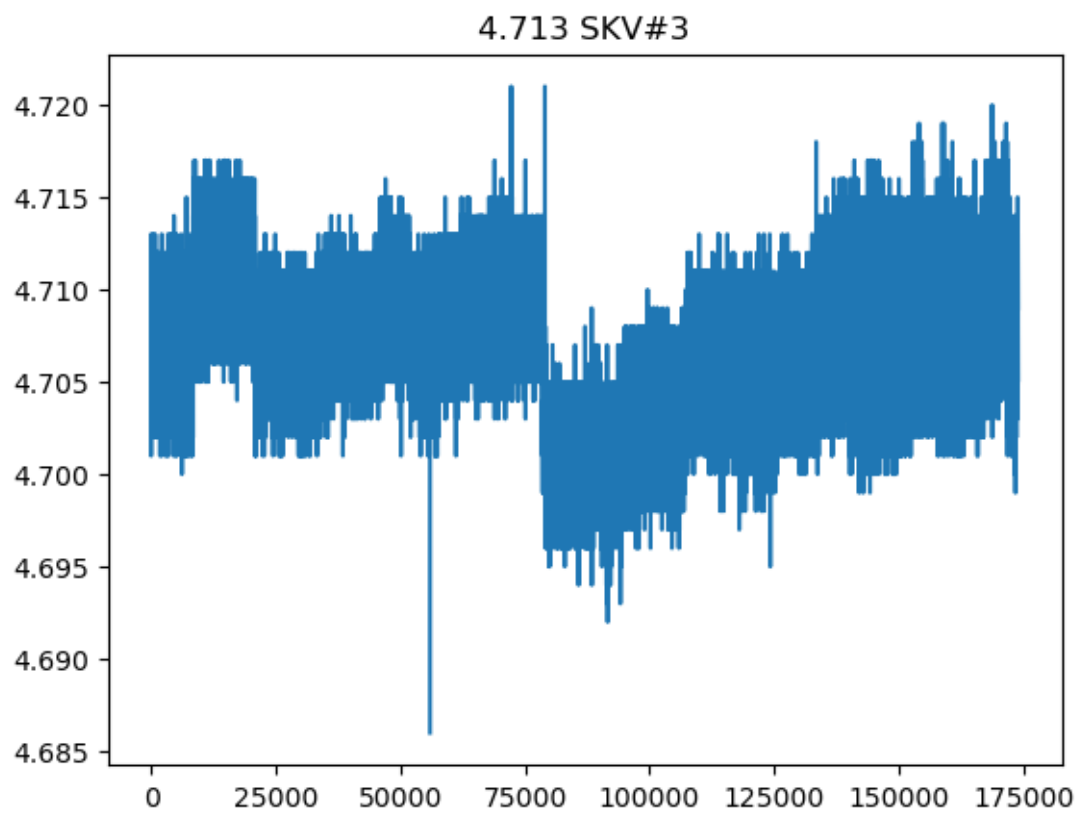


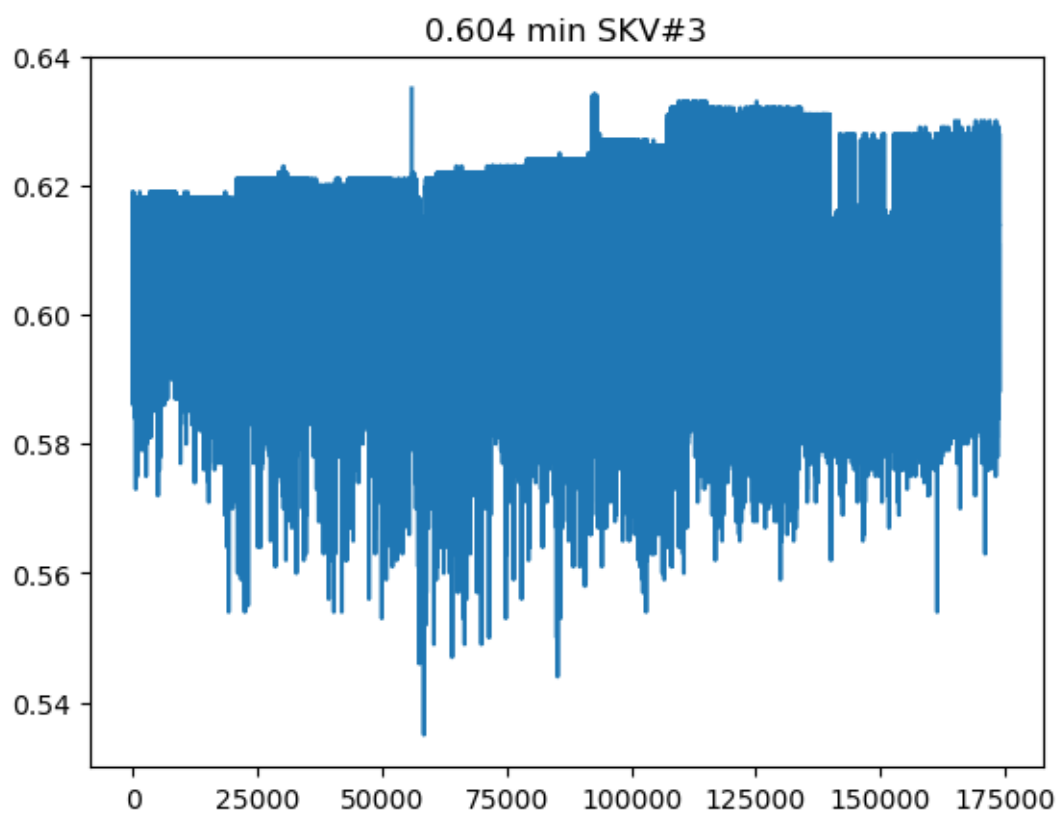


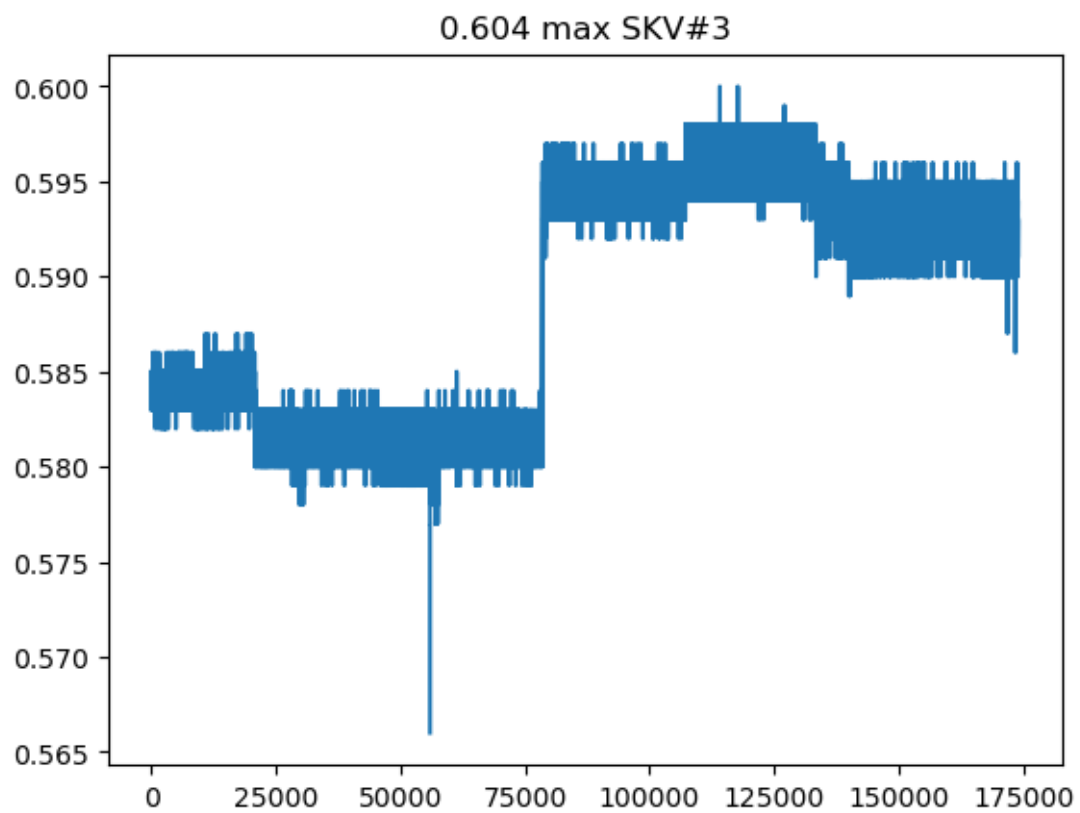


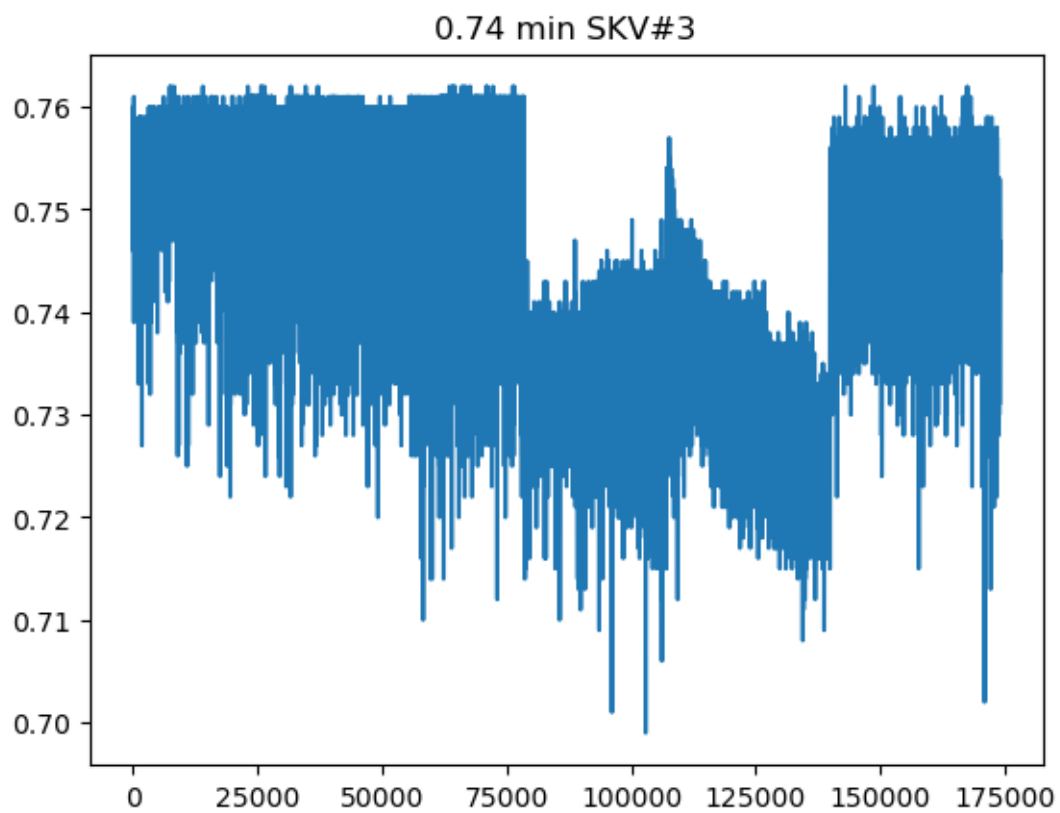


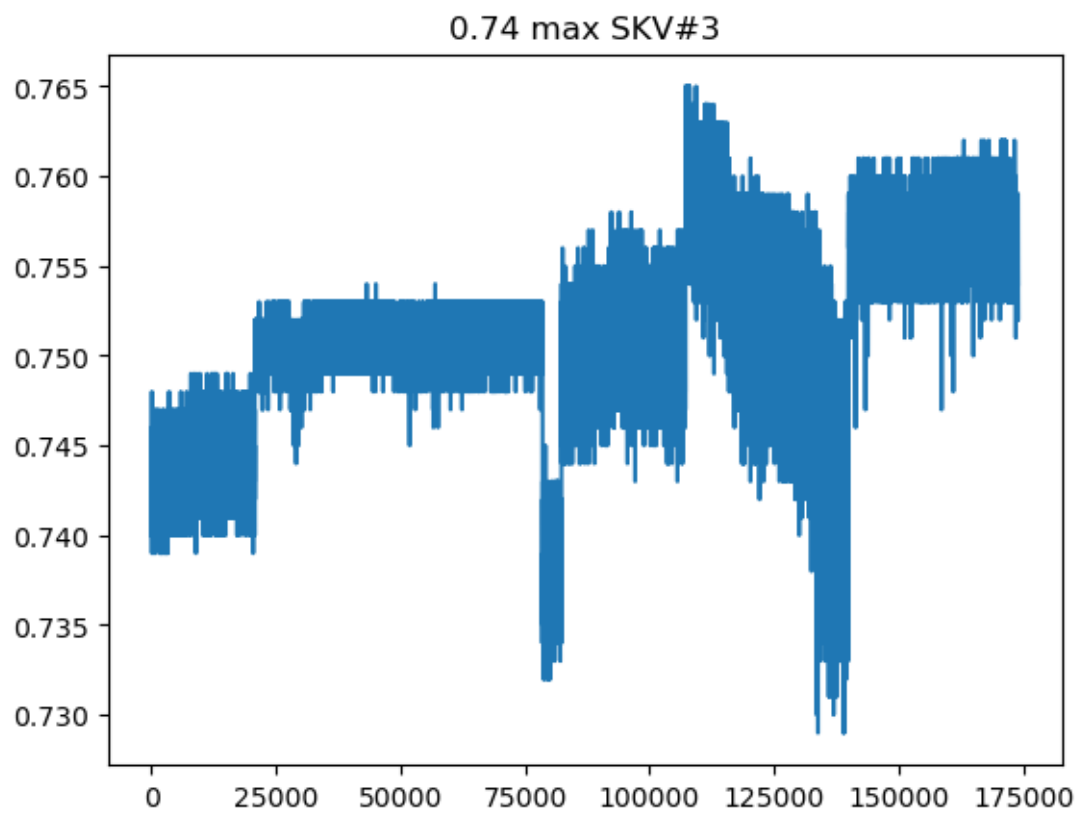


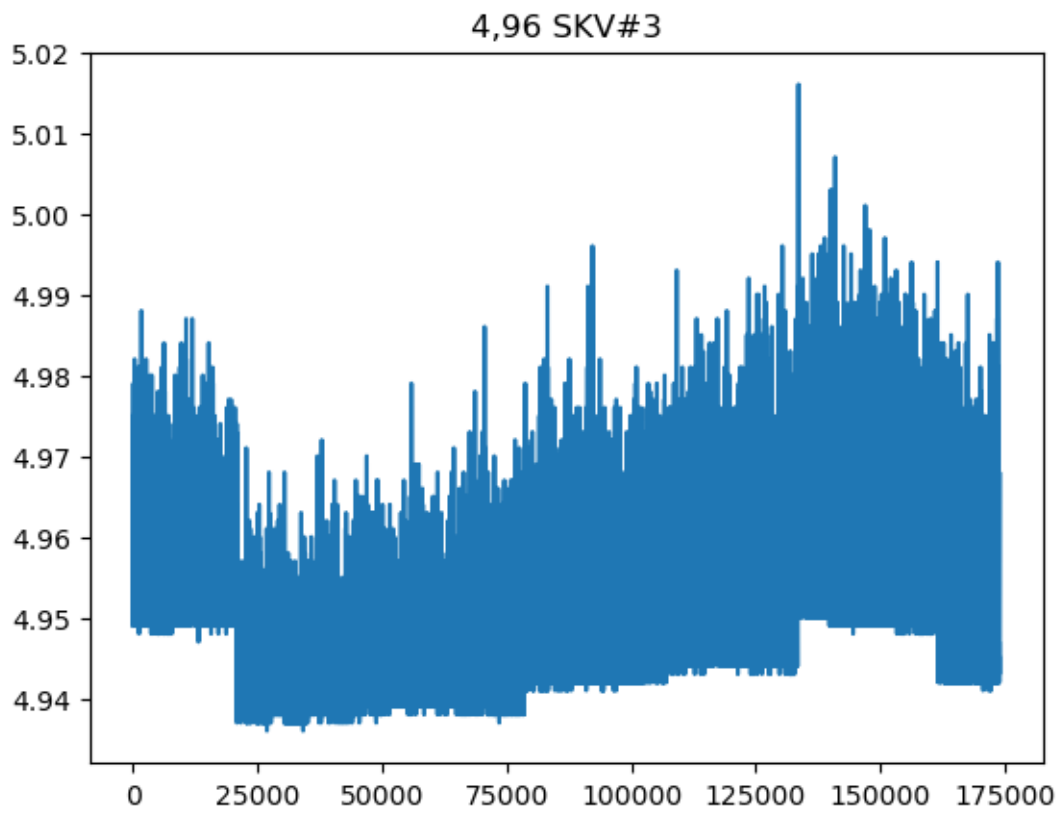


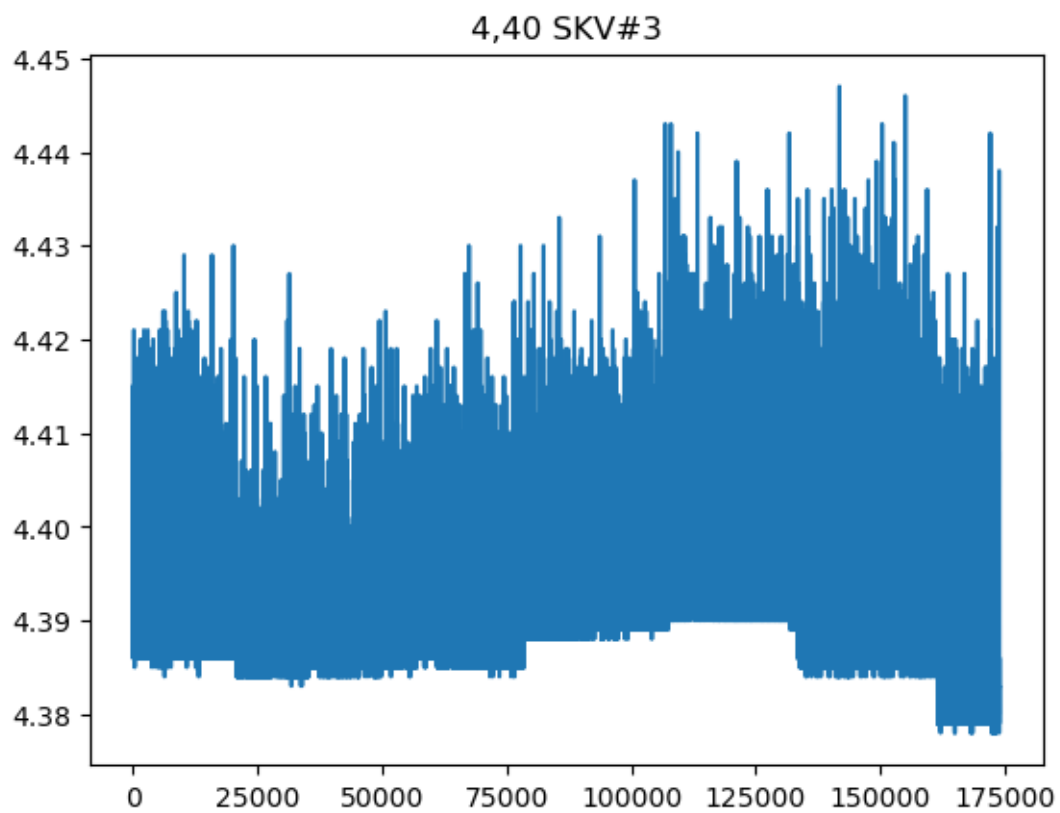




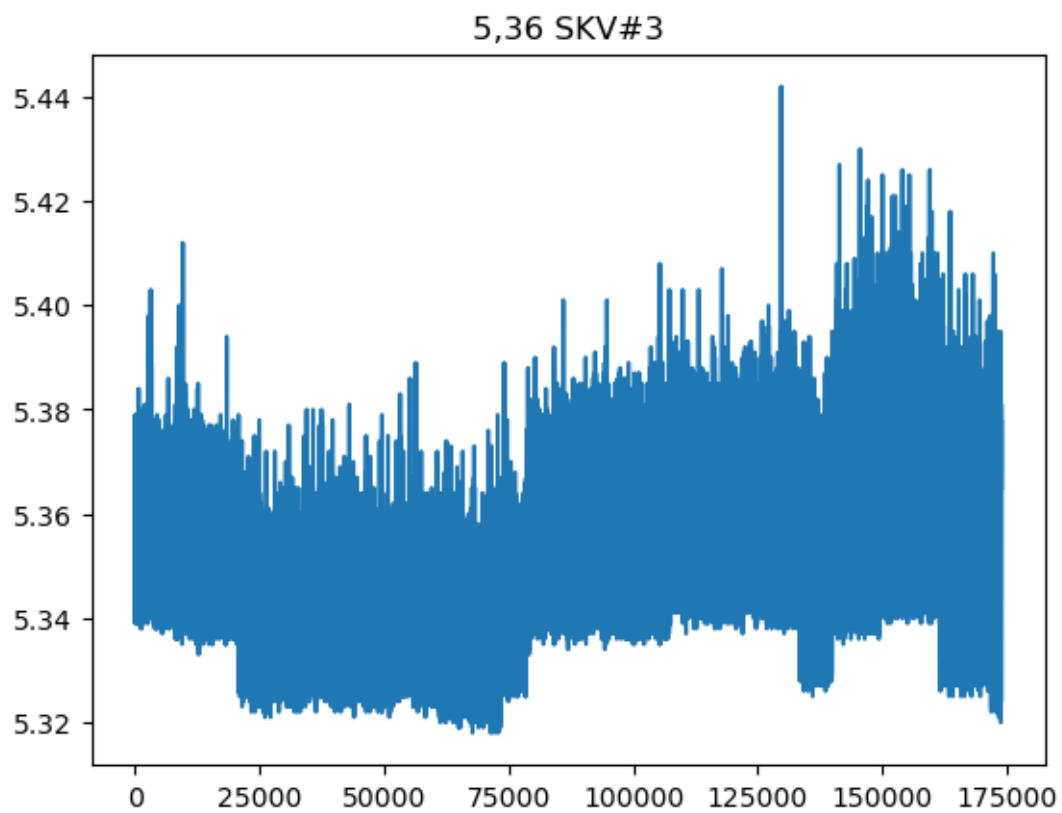


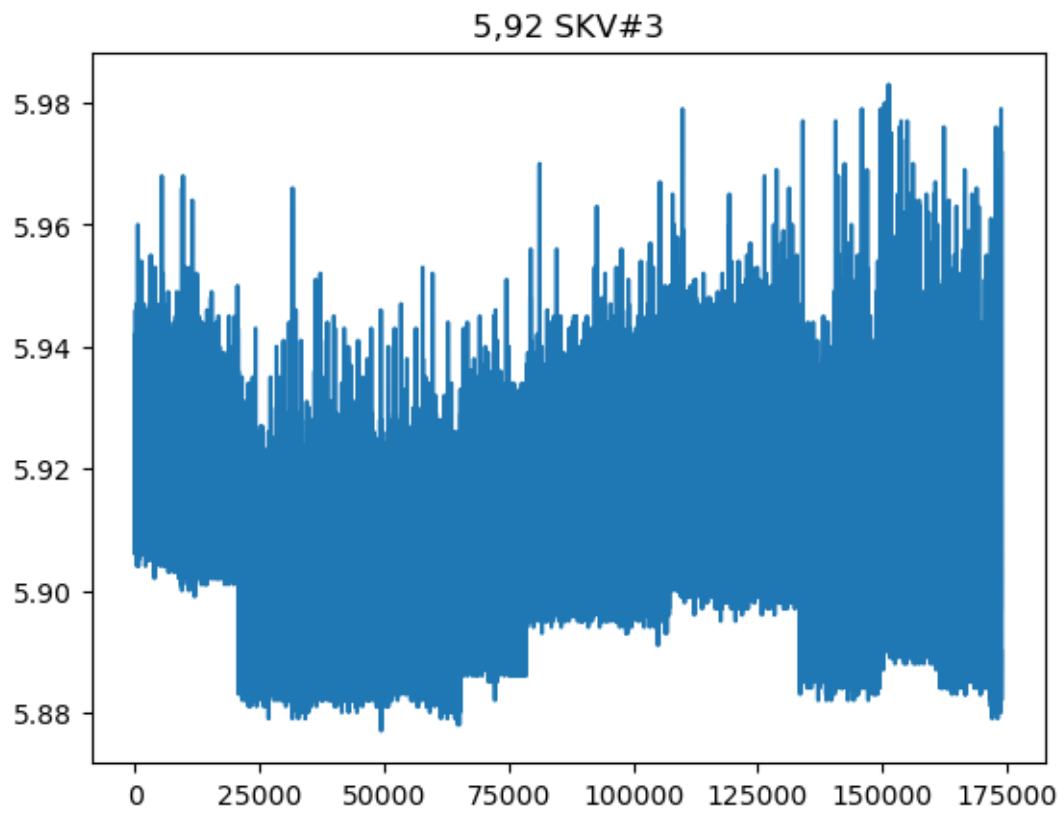


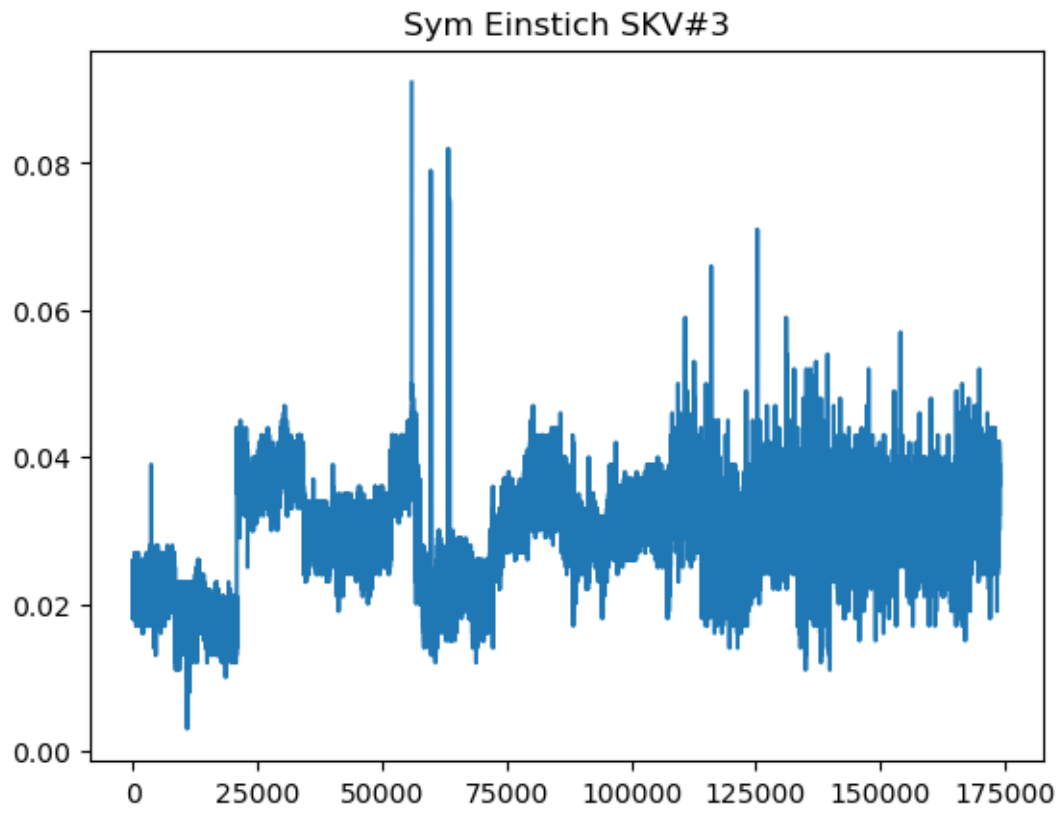


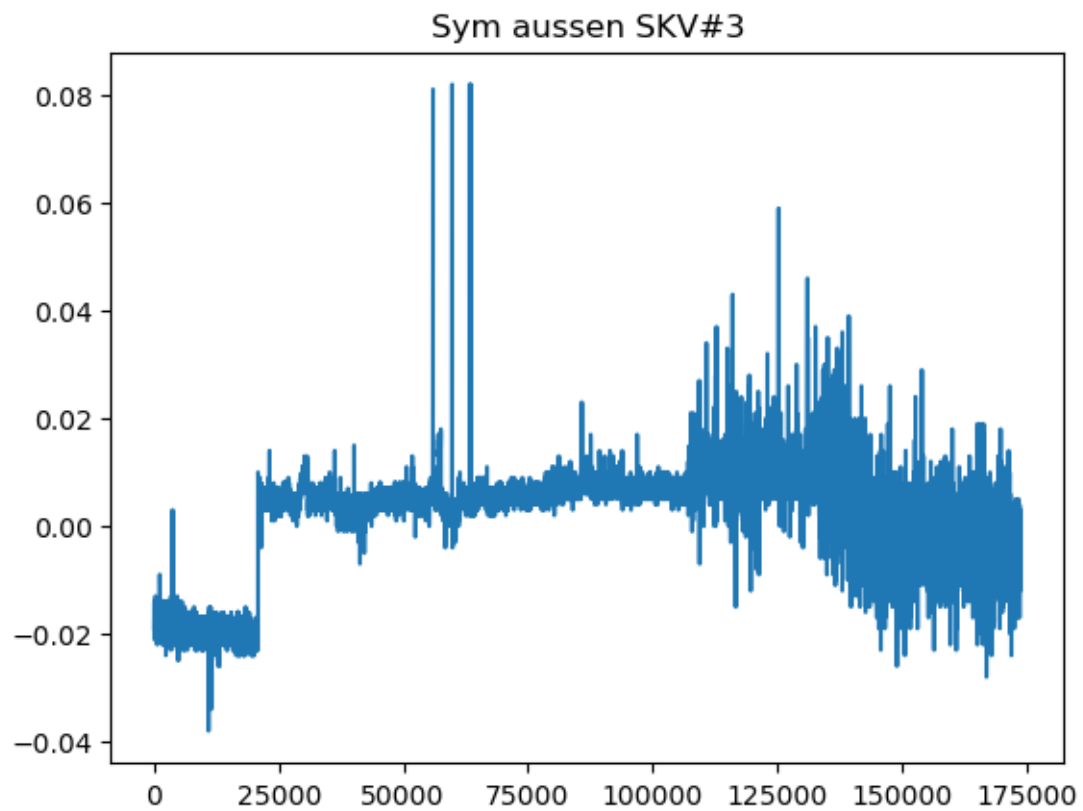


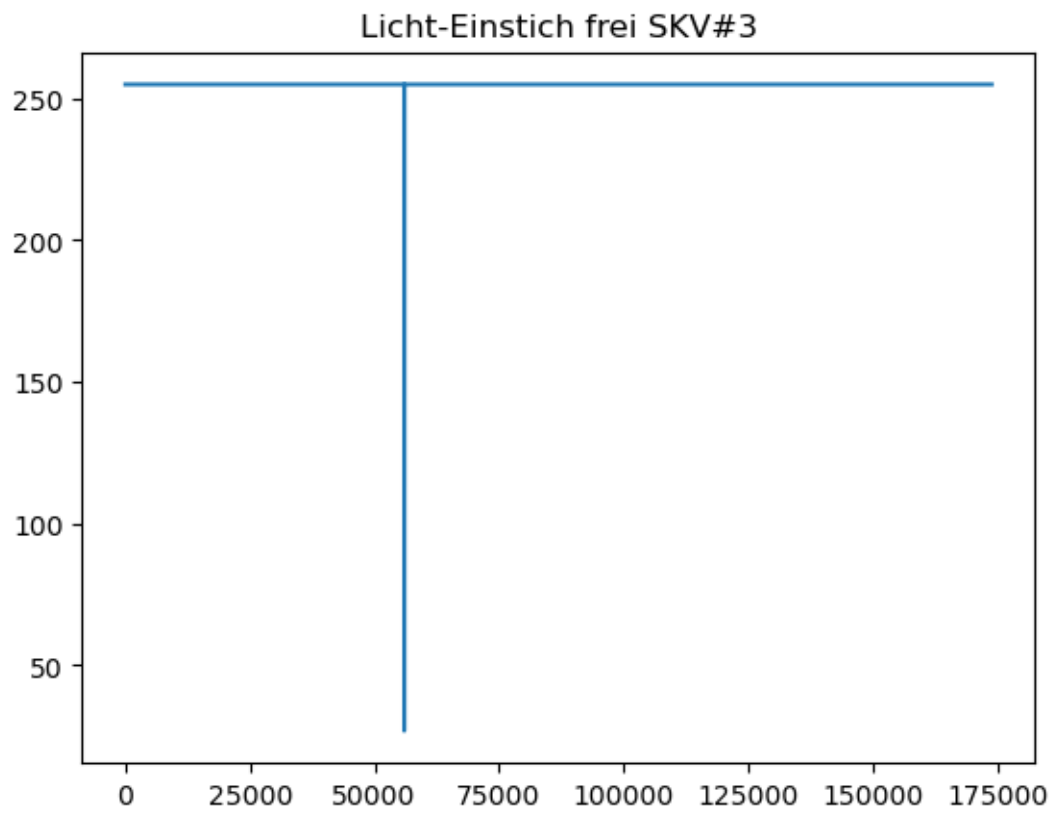


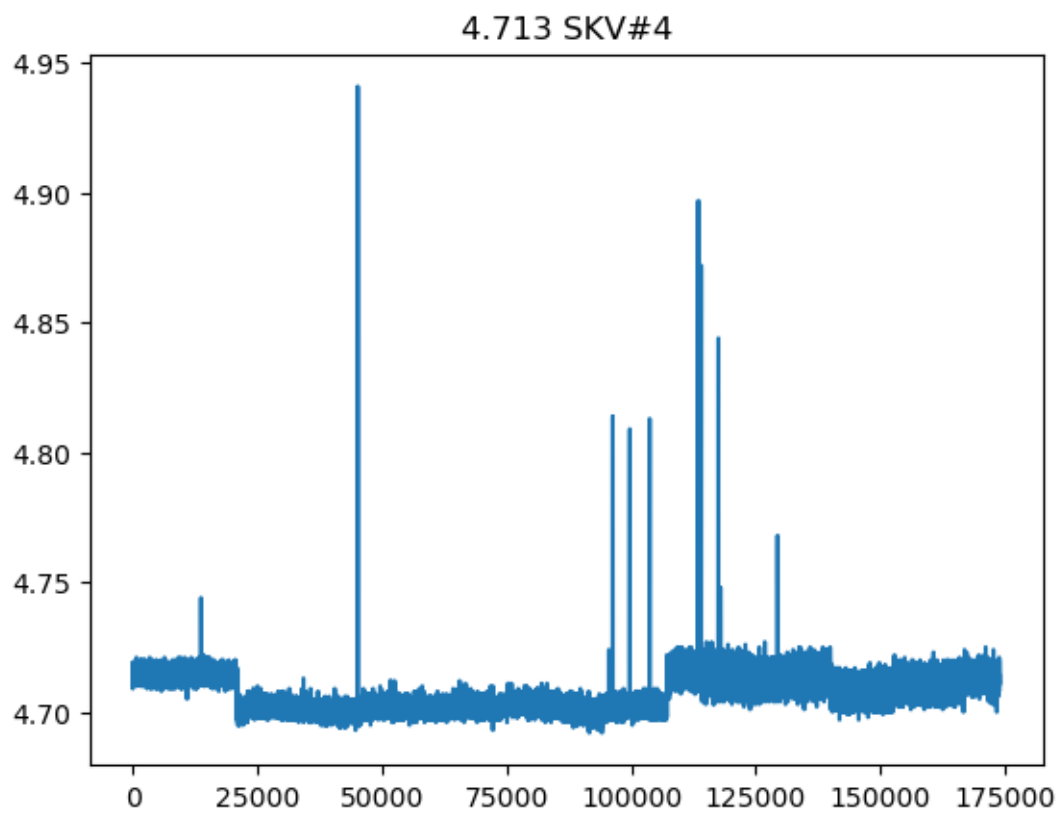


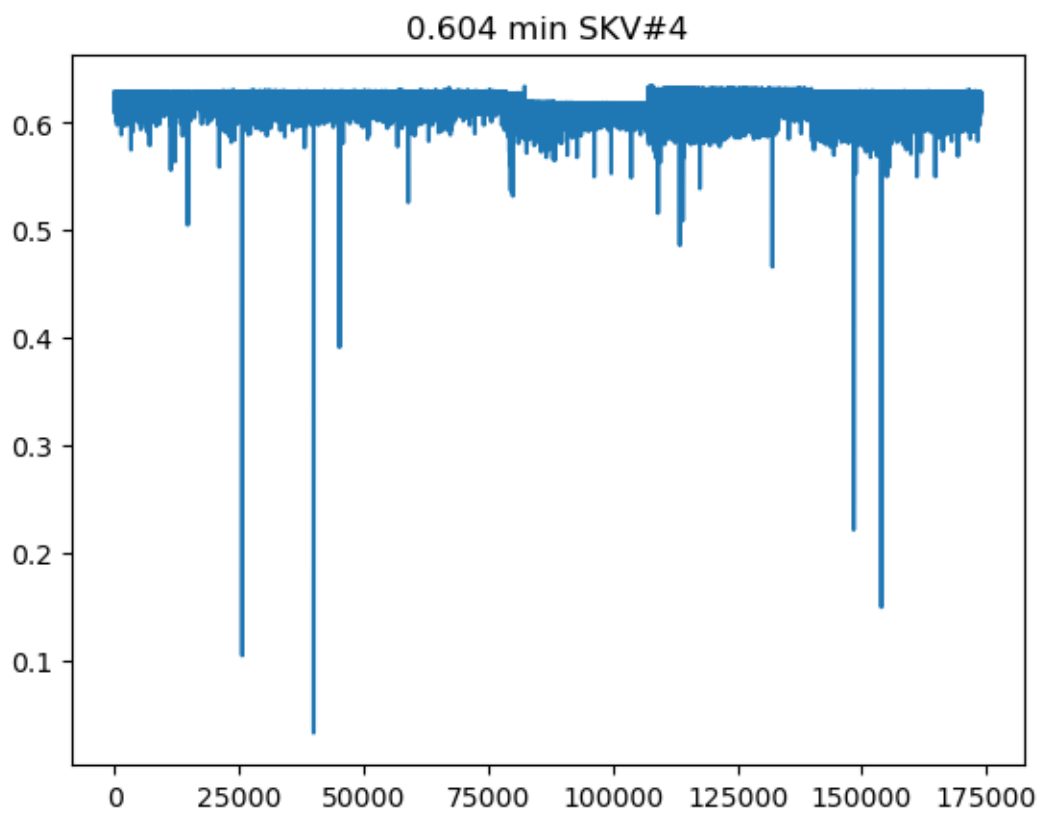


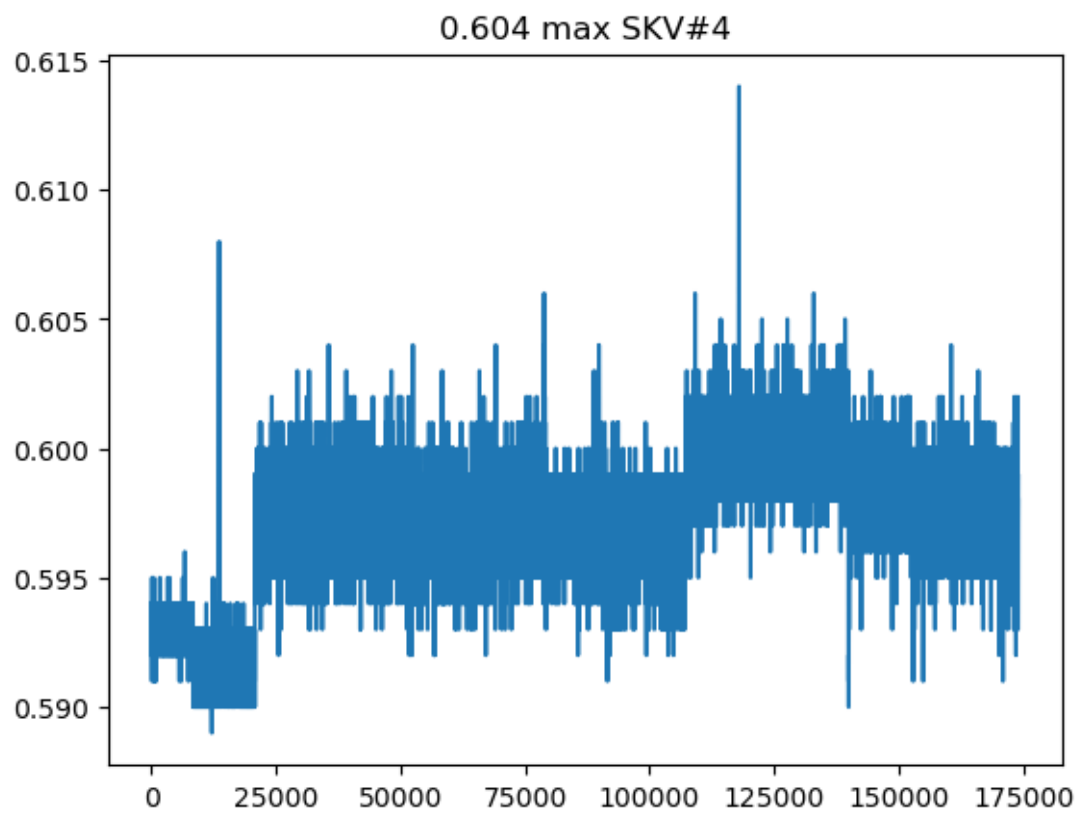




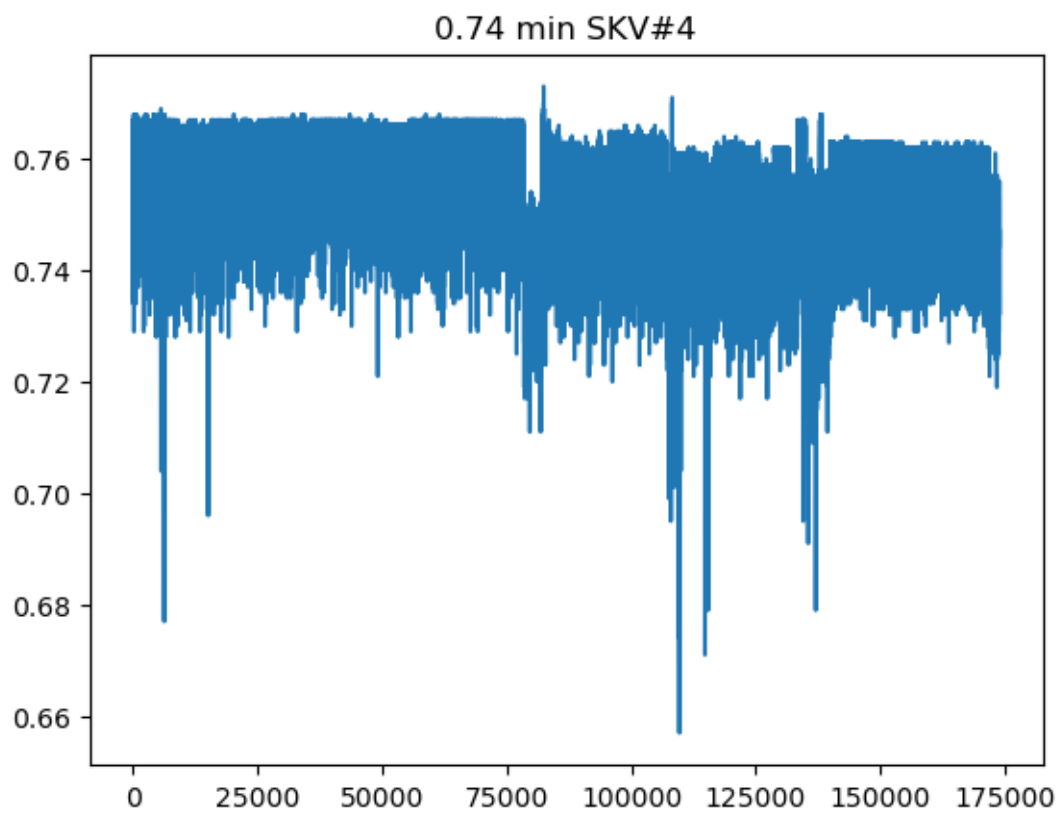


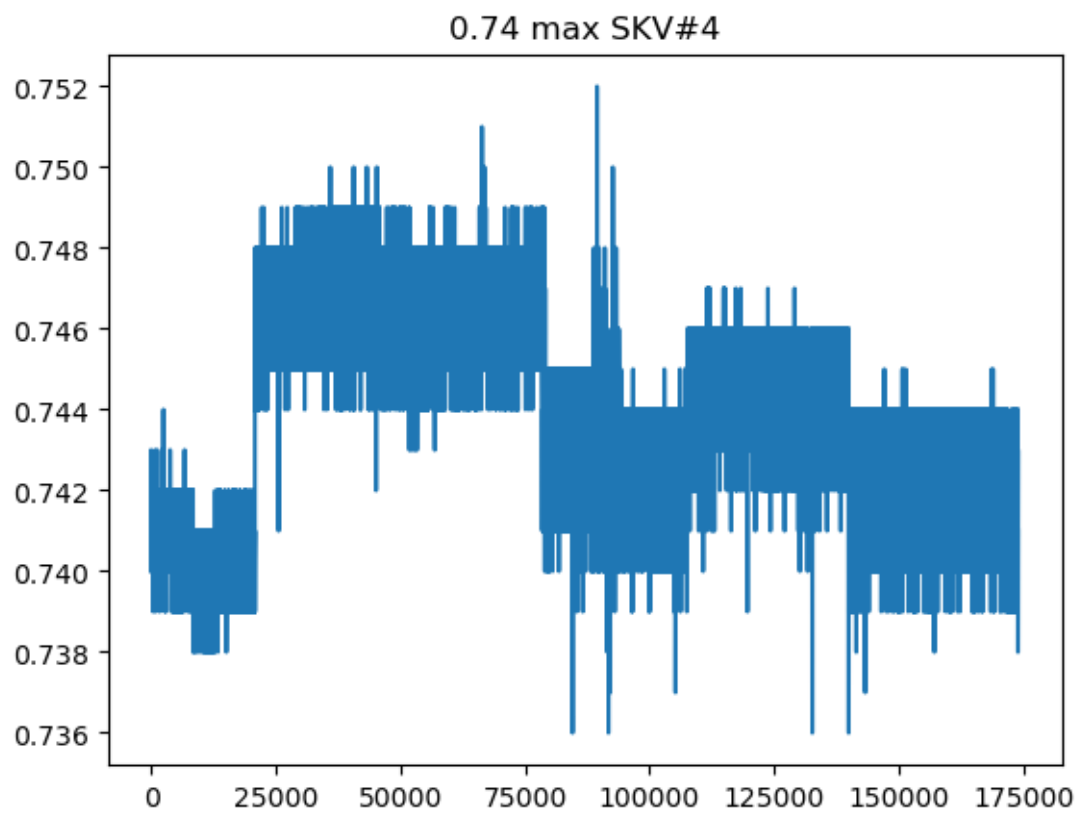


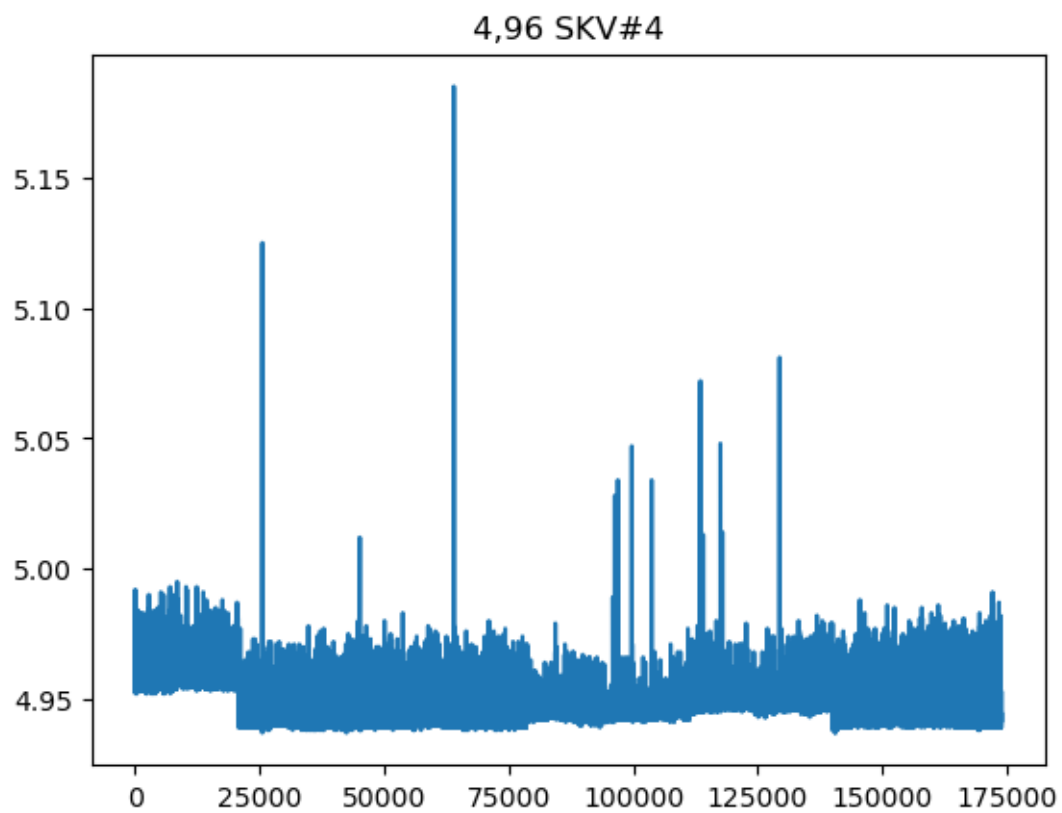


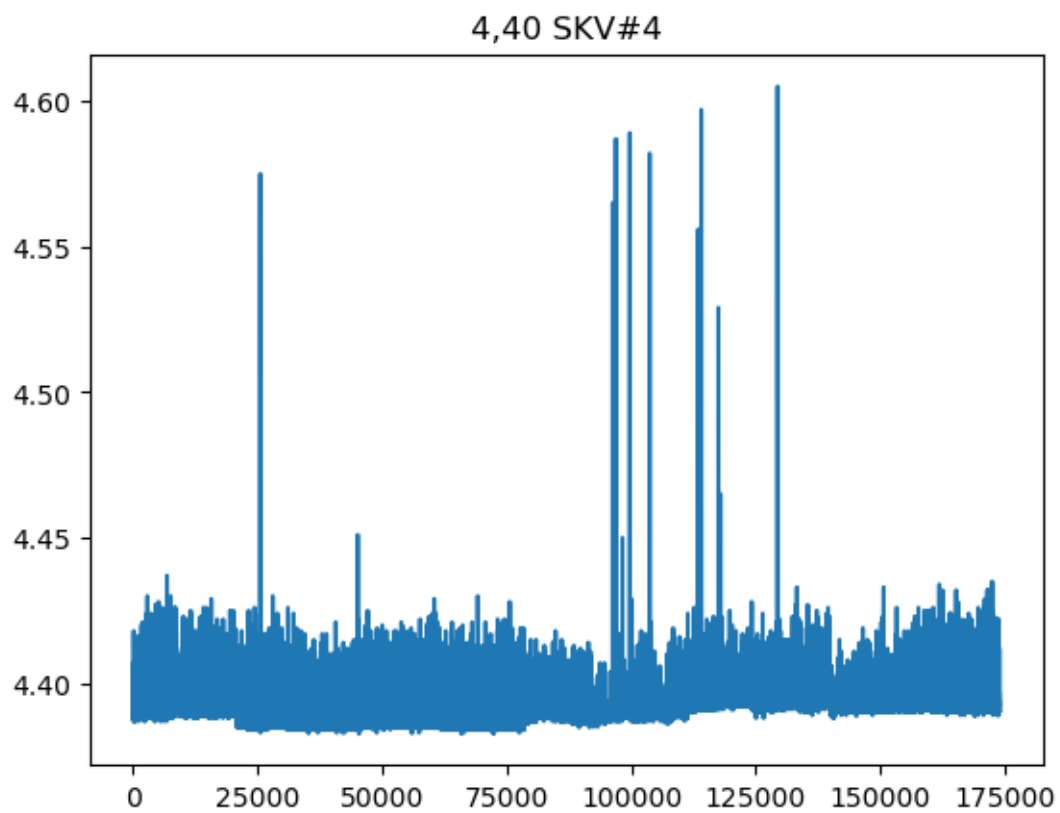


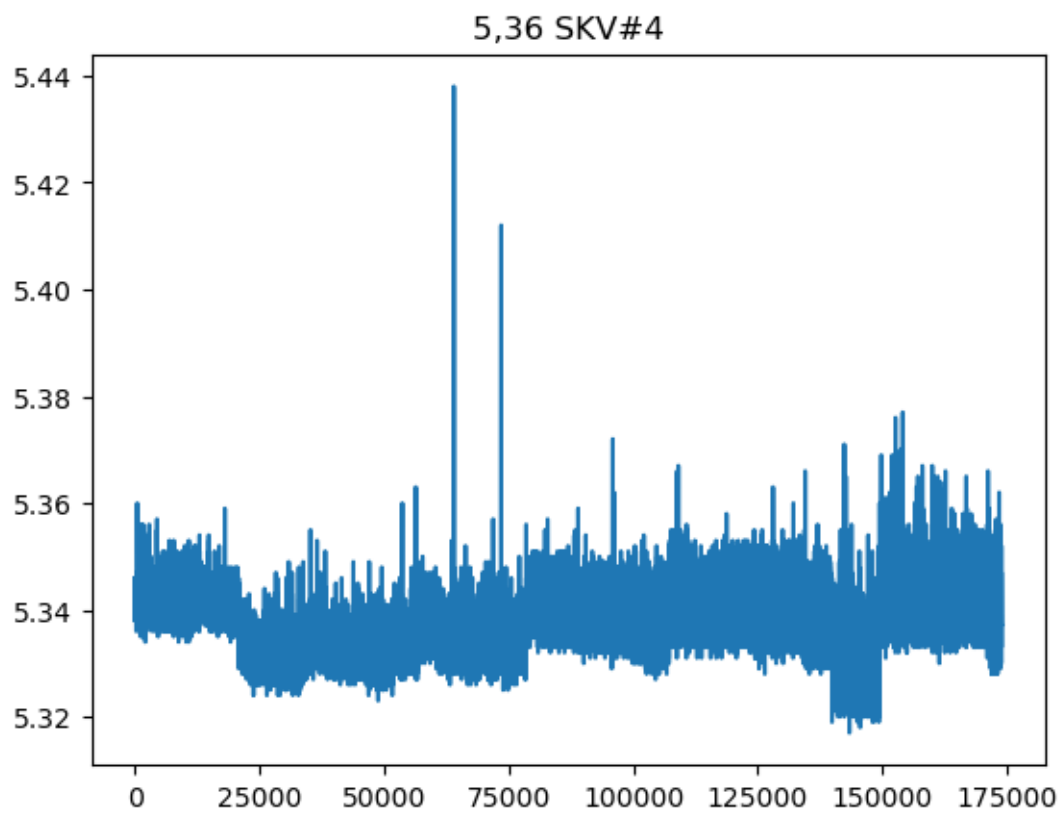


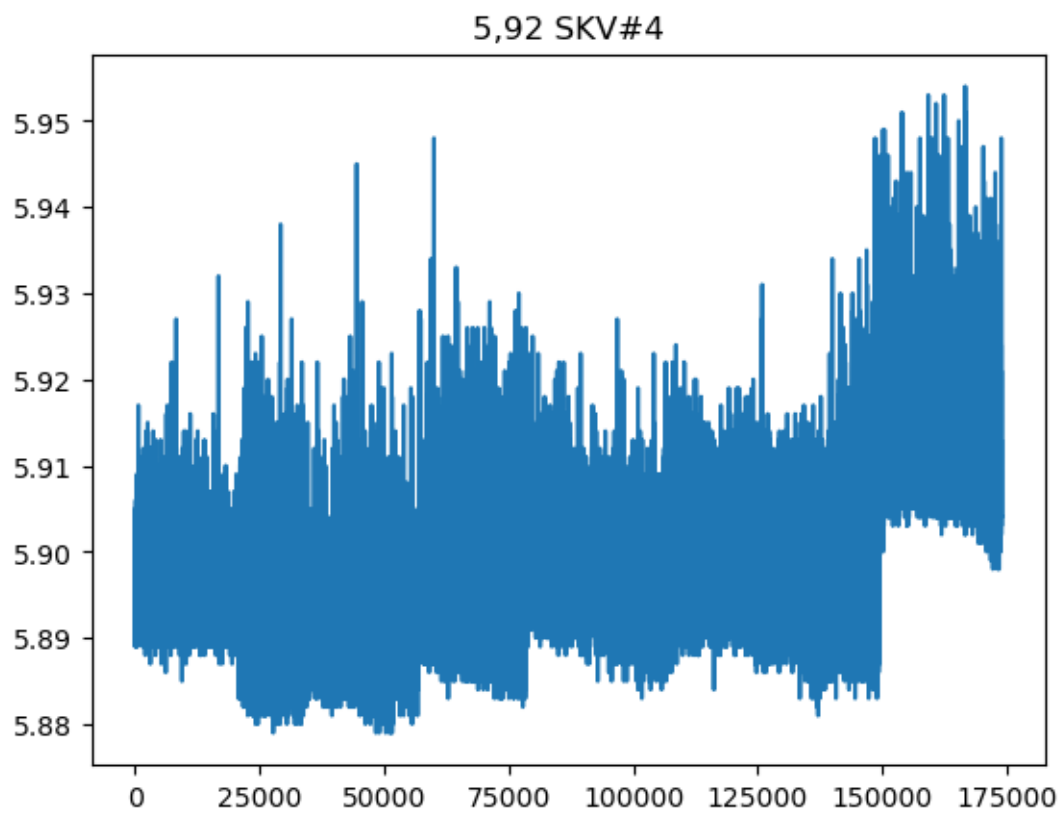


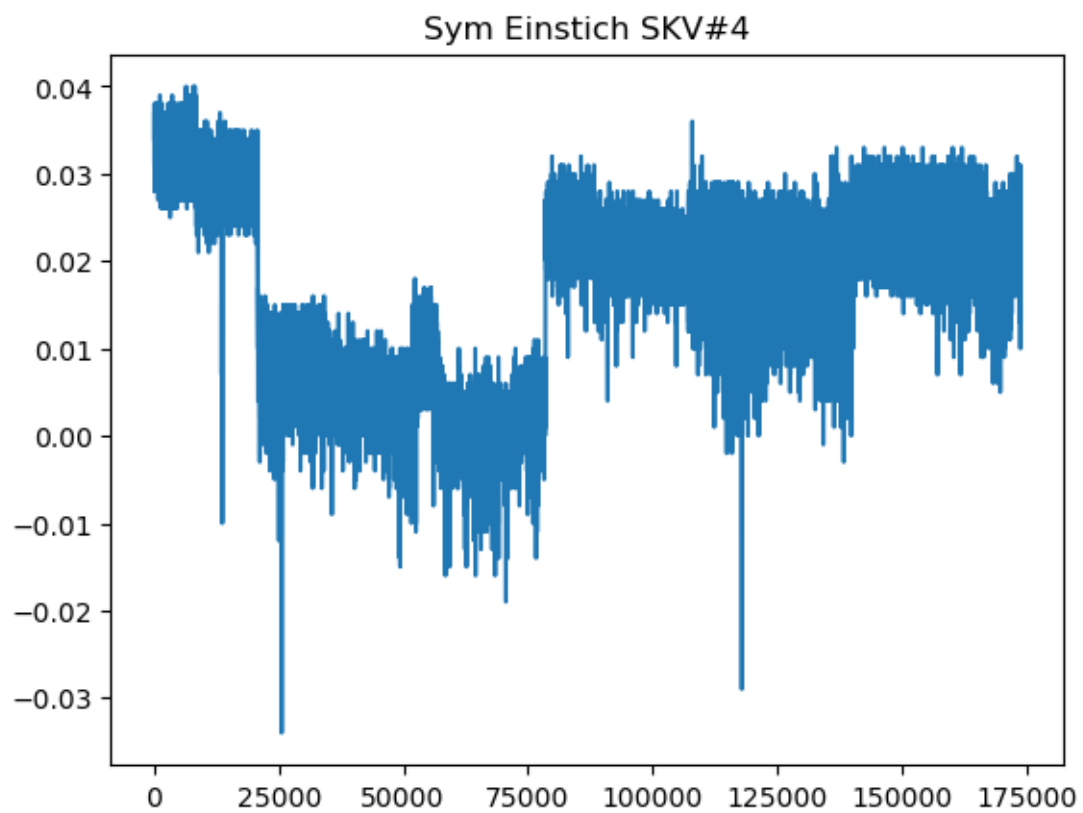


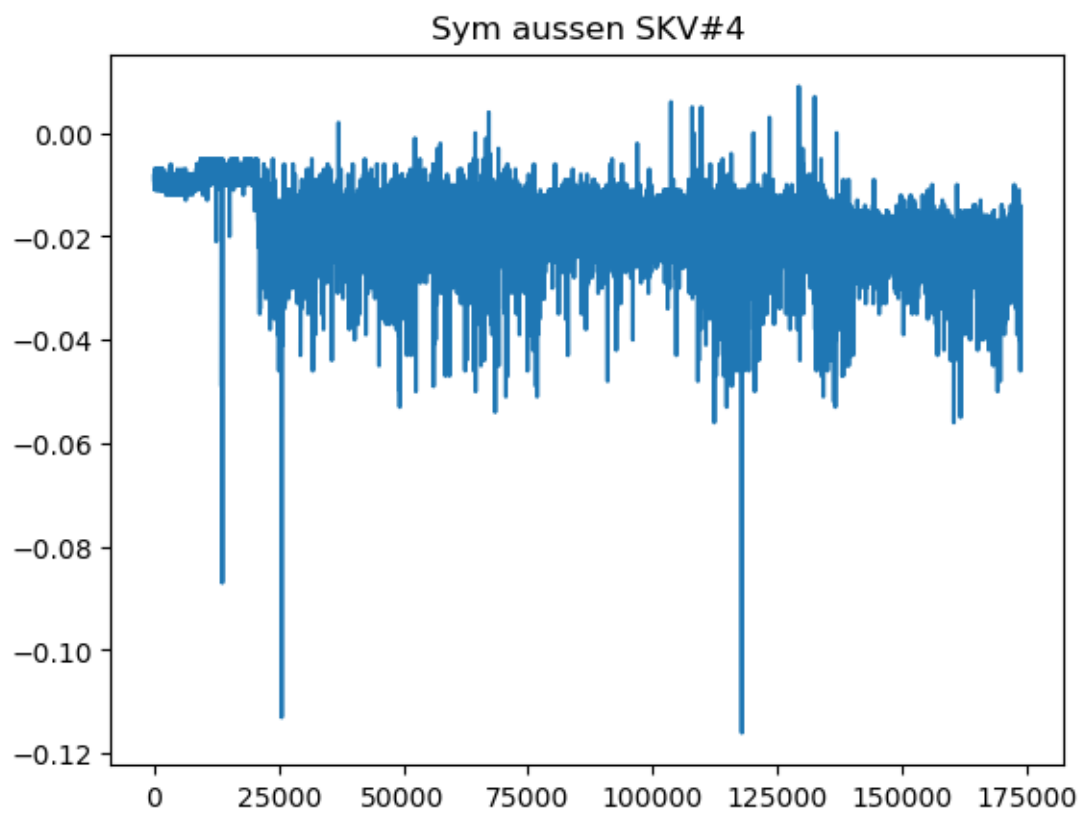




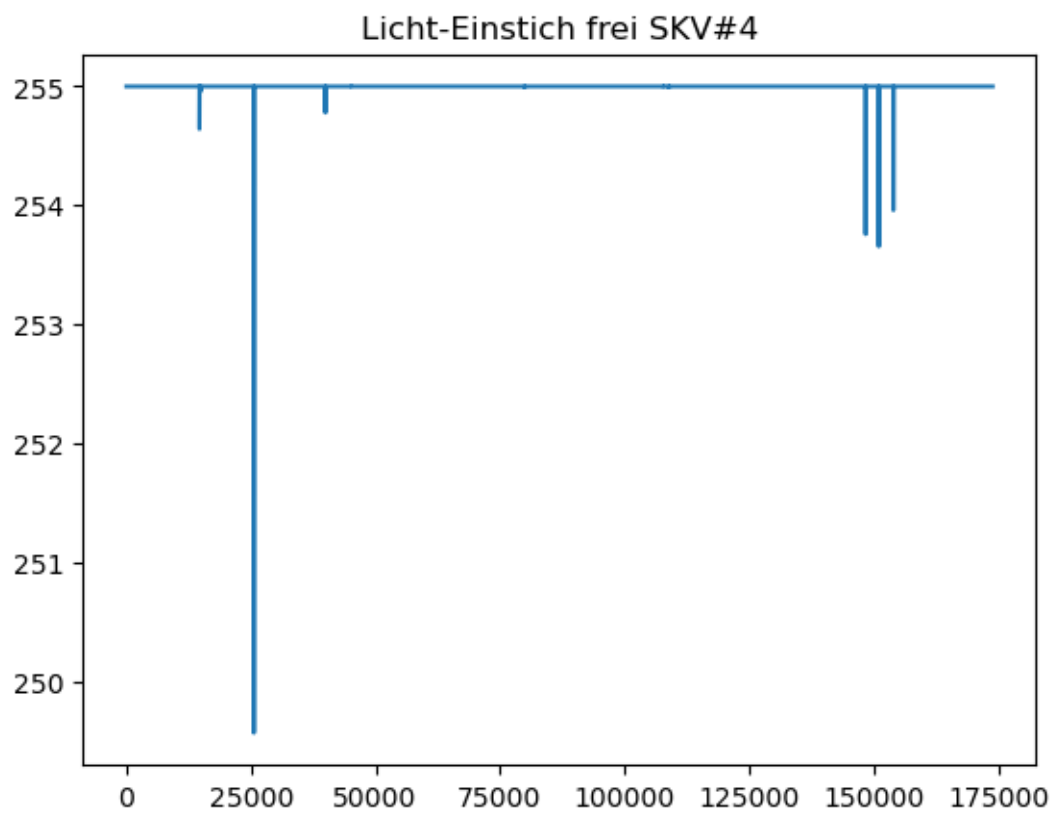


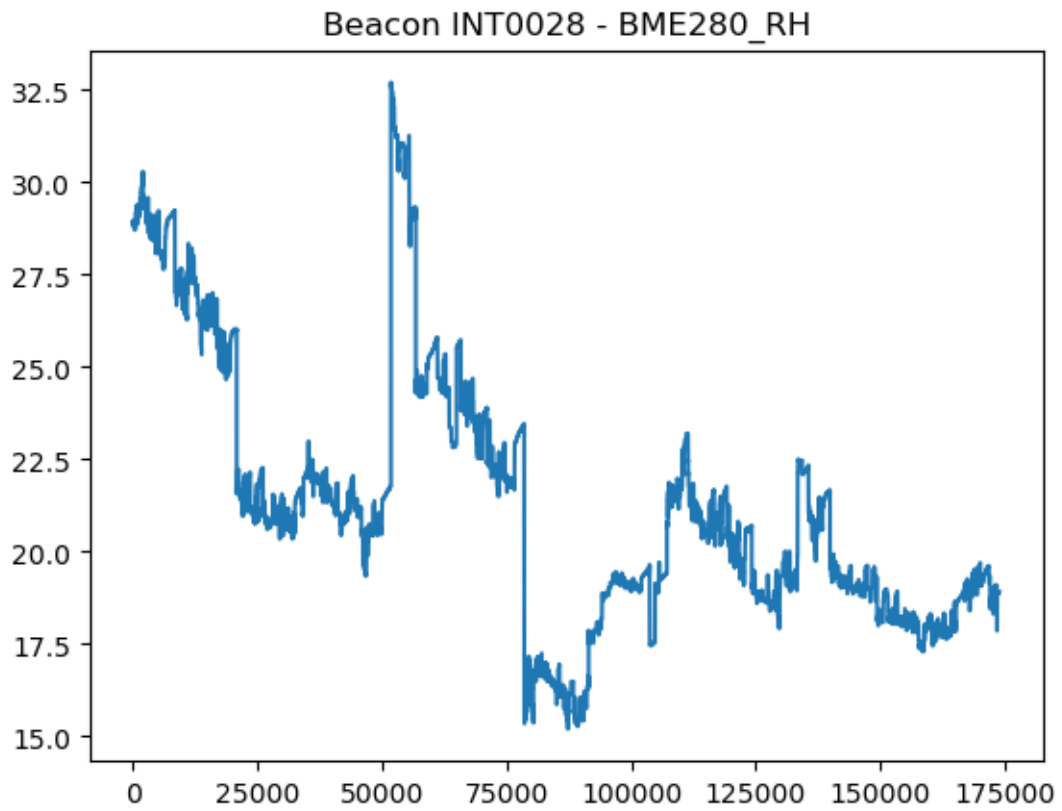


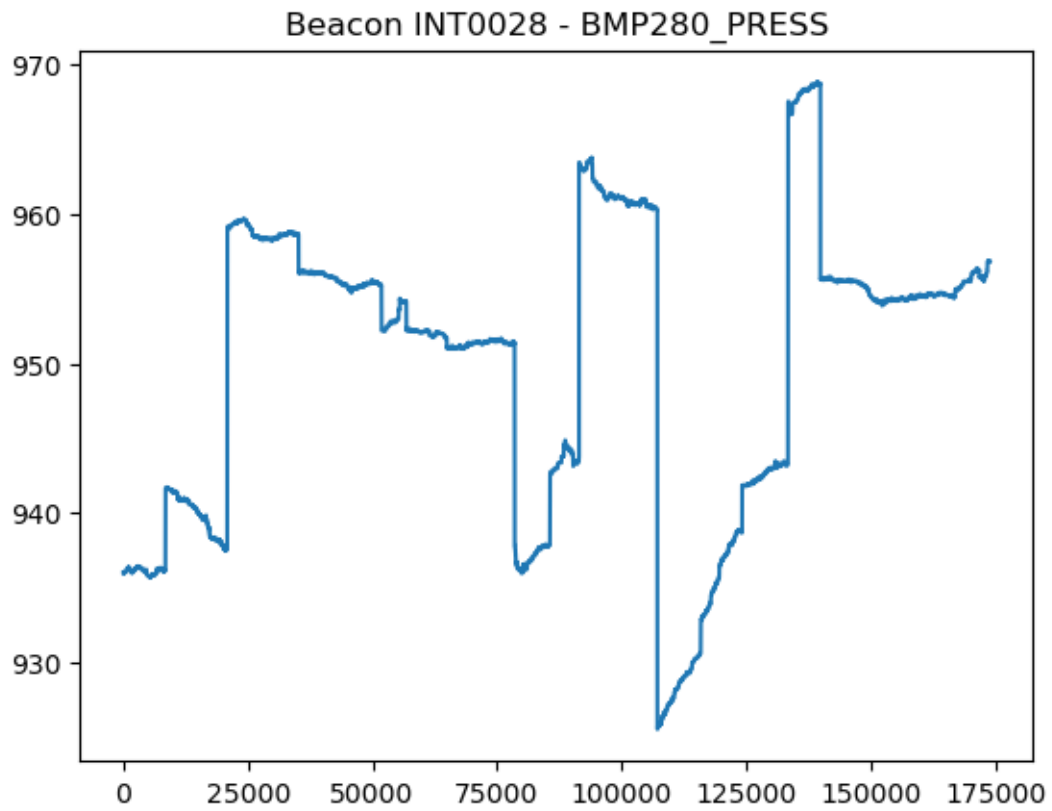


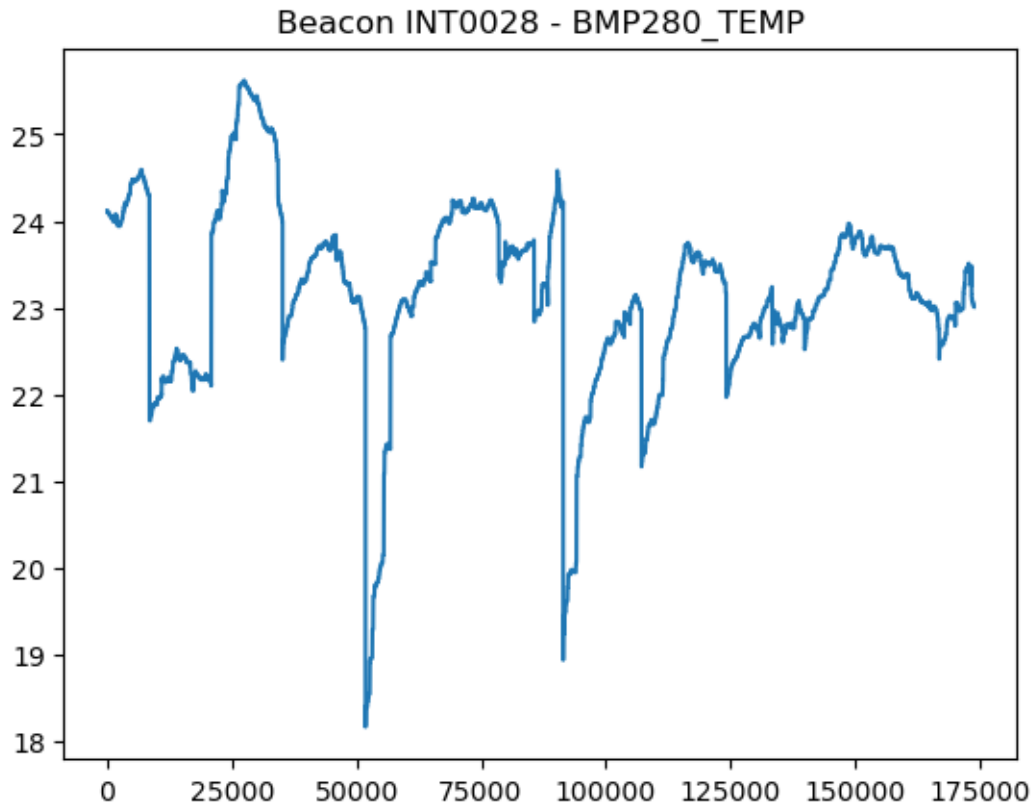












```
[9]: for f in interestingColumns:
      na = X[f].isna()
      toDrop = X[na].index
      print(f"{f} dropped {toDrop}")
      X = X.drop(toDrop)

      #Xna = X[na > 0].drop()
```

```
ST_CH1_P_F0 dropped Index([], dtype='int64')
ST_CH1_P_F1 dropped Index([], dtype='int64')
ST_CH1_P_F2 dropped Index([], dtype='int64')
ST_CH1_P_F3 dropped Index([], dtype='int64')
ST_CH1_P_F4 dropped Index([], dtype='int64')
ST_CH1_P_F5 dropped Index([], dtype='int64')
ST_CH1_P_F6 dropped Index([], dtype='int64')
ST_CH1_P_F7 dropped Index([], dtype='int64')
ST_CH1_P_F8 dropped Index([], dtype='int64')
ST_CH1_P_F9 dropped Index([], dtype='int64')
ST_CH1_P_F10 dropped Index([], dtype='int64')
ST_CH1_P_F11 dropped Index([], dtype='int64')
ST_CH1_P0_F0 dropped Index([], dtype='int64')
```

```

ST_CH1_P0_F1 dropped Index([], dtype='int64')
ST_CH1_P0_F2 dropped Index([], dtype='int64')
ST_CH1_P0_F3 dropped Index([], dtype='int64')
ST_CH1_P0_F4 dropped Index([], dtype='int64')
ST_CH1_P0_F5 dropped Index([], dtype='int64')
ST_CH1_P0_F6 dropped Index([], dtype='int64')
ST_CH1_P0_F7 dropped Index([], dtype='int64')
ST_CH1_P0_F8 dropped Index([], dtype='int64')
ST_CH1_P0_F9 dropped Index([], dtype='int64')
ST_CH1_P0_F10 dropped Index([], dtype='int64')
ST_CH1_P0_F11 dropped Index([], dtype='int64')
ST_CH1_P1_F0 dropped Index([], dtype='int64')
ST_CH1_P1_F1 dropped Index([], dtype='int64')
ST_CH1_P1_F2 dropped Index([], dtype='int64')
ST_CH1_P1_F3 dropped Index([], dtype='int64')
ST_CH1_P1_F4 dropped Index([], dtype='int64')
ST_CH1_P1_F5 dropped Index([], dtype='int64')
ST_CH1_P1_F6 dropped Index([], dtype='int64')
ST_CH1_P1_F7 dropped Index([], dtype='int64')
ST_CH1_P1_F8 dropped Index([], dtype='int64')
ST_CH1_P1_F9 dropped Index([], dtype='int64')
ST_CH1_P1_F10 dropped Index([], dtype='int64')
ST_CH1_P1_F11 dropped Index([], dtype='int64')
ST_CH1_P2_F0 dropped Index([], dtype='int64')
ST_CH1_P2_F1 dropped Index([], dtype='int64')
ST_CH1_P2_F2 dropped Index([], dtype='int64')
ST_CH1_P2_F3 dropped Index([], dtype='int64')
ST_CH1_P2_F4 dropped Index([], dtype='int64')
ST_CH1_P2_F5 dropped Index([], dtype='int64')
ST_CH1_P2_F6 dropped Index([], dtype='int64')
ST_CH1_P2_F7 dropped Index([], dtype='int64')
ST_CH1_P2_F8 dropped Index([], dtype='int64')
ST_CH1_P2_F9 dropped Index([], dtype='int64')
ST_CH1_P2_F10 dropped Index([], dtype='int64')
ST_CH1_P2_F11 dropped Index([], dtype='int64')
ST_CH2_P_F0 dropped Index([], dtype='int64')
ST_CH2_P_F1 dropped Index([], dtype='int64')
ST_CH2_P_F2 dropped Index([], dtype='int64')
ST_CH2_P_F3 dropped Index([], dtype='int64')
ST_CH2_P_F4 dropped Index([], dtype='int64')
ST_CH2_P_F5 dropped Index([], dtype='int64')
ST_CH2_P_F6 dropped Index([], dtype='int64')
ST_CH2_P_F7 dropped Index([], dtype='int64')
ST_CH2_P_F8 dropped Index([], dtype='int64')
ST_CH2_P_F9 dropped Index([], dtype='int64')
ST_CH2_P_F10 dropped Index([], dtype='int64')
ST_CH2_P_F11 dropped Index([], dtype='int64')
ST_CH2_P0_F0 dropped Index([], dtype='int64')

```

```

ST_CH2_P0_F1 dropped Index([], dtype='int64')
ST_CH2_P0_F2 dropped Index([], dtype='int64')
ST_CH2_P0_F3 dropped Index([], dtype='int64')
ST_CH2_P0_F4 dropped Index([], dtype='int64')
ST_CH2_P0_F5 dropped Index([], dtype='int64')
ST_CH2_P0_F6 dropped Index([], dtype='int64')
ST_CH2_P0_F7 dropped Index([], dtype='int64')
ST_CH2_P0_F8 dropped Index([], dtype='int64')
ST_CH2_P0_F9 dropped Index([], dtype='int64')
ST_CH2_P0_F10 dropped Index([], dtype='int64')
ST_CH2_P0_F11 dropped Index([], dtype='int64')
ST_CH2_P1_F0 dropped Index([], dtype='int64')
ST_CH2_P1_F1 dropped Index([], dtype='int64')
ST_CH2_P1_F2 dropped Index([], dtype='int64')
ST_CH2_P1_F3 dropped Index([], dtype='int64')
ST_CH2_P1_F4 dropped Index([], dtype='int64')
ST_CH2_P1_F5 dropped Index([], dtype='int64')
ST_CH2_P1_F6 dropped Index([], dtype='int64')
ST_CH2_P1_F7 dropped Index([], dtype='int64')
ST_CH2_P1_F8 dropped Index([], dtype='int64')
ST_CH2_P1_F9 dropped Index([], dtype='int64')
ST_CH2_P1_F10 dropped Index([], dtype='int64')
ST_CH2_P1_F11 dropped Index([], dtype='int64')
ST_CH2_P2_F0 dropped Index([], dtype='int64')
ST_CH2_P2_F1 dropped Index([], dtype='int64')
ST_CH2_P2_F2 dropped Index([], dtype='int64')
ST_CH2_P2_F3 dropped Index([], dtype='int64')
ST_CH2_P2_F4 dropped Index([], dtype='int64')
ST_CH2_P2_F5 dropped Index([], dtype='int64')
ST_CH2_P2_F6 dropped Index([], dtype='int64')
ST_CH2_P2_F7 dropped Index([], dtype='int64')
ST_CH2_P2_F8 dropped Index([], dtype='int64')
ST_CH2_P2_F9 dropped Index([], dtype='int64')
ST_CH2_P2_F10 dropped Index([], dtype='int64')
ST_CH2_P2_F11 dropped Index([], dtype='int64')
4.713 SKV#1 dropped Index([], dtype='int64')
0.604 min SKV#1 dropped Index([67529], dtype='int64')
0.604 max SKV#1 dropped Index([], dtype='int64')
0.74 min SKV#1 dropped Index([], dtype='int64')
0.74 max SKV#1 dropped Index([], dtype='int64')
4,96 SKV#1 dropped Index([], dtype='int64')
4,40 SKV#1 dropped Index([], dtype='int64')
5,36 SKV#1 dropped Index([], dtype='int64')
5,92 SKV#1 dropped Index([], dtype='int64')
Sym aussen SKV#1 dropped Index([], dtype='int64')
Sym Einstich SKV#1 dropped Index([], dtype='int64')
Licht-Einstich frei SKV#1 dropped Index([], dtype='int64')
4.713 SKV#2 dropped Index([15807, 15808, 15809, 15810, 15811, 15812, 15813,

```

```

15900, 16556, 17224,
    18655, 19591, 19649, 20690, 20694, 82760],
    dtype='int64')
0.604 min SKV#2 dropped Index([], dtype='int64')
0.604 max SKV#2 dropped Index([], dtype='int64')
0.74 min SKV#2 dropped Index([], dtype='int64')
0.74 max SKV#2 dropped Index([], dtype='int64')
4,96 SKV#2 dropped Index([], dtype='int64')
4,40 SKV#2 dropped Index([], dtype='int64')
5,36 SKV#2 dropped Index([], dtype='int64')
5,92 SKV#2 dropped Index([], dtype='int64')
Sym Einstich SKV#2 dropped Index([], dtype='int64')
Sym aussen SKV#2 dropped Index([], dtype='int64')
Licht-Einstich frei SKV#2 dropped Index([], dtype='int64')
4.713 SKV#3 dropped Index([55958, 55959, 55960, 55961, 55962, 55963, 55964,
55965, 55966, 55967,
    55968, 55969, 55970],
    dtype='int64')
0.604 min SKV#3 dropped Index([], dtype='int64')
0.604 max SKV#3 dropped Index([], dtype='int64')
0.74 min SKV#3 dropped Index([], dtype='int64')
0.74 max SKV#3 dropped Index([], dtype='int64')
4,96 SKV#3 dropped Index([], dtype='int64')
4,40 SKV#3 dropped Index([], dtype='int64')
5,36 SKV#3 dropped Index([], dtype='int64')
5,92 SKV#3 dropped Index([], dtype='int64')
Sym Einstich SKV#3 dropped Index([], dtype='int64')
Sym aussen SKV#3 dropped Index([], dtype='int64')
Licht-Einstich frei SKV#3 dropped Index([], dtype='int64')
4.713 SKV#4 dropped Index([25593], dtype='int64')
0.604 min SKV#4 dropped Index([], dtype='int64')
0.604 max SKV#4 dropped Index([], dtype='int64')
0.74 min SKV#4 dropped Index([], dtype='int64')
0.74 max SKV#4 dropped Index([], dtype='int64')
4,96 SKV#4 dropped Index([], dtype='int64')
4,40 SKV#4 dropped Index([35250], dtype='int64')
5,36 SKV#4 dropped Index([], dtype='int64')
5,92 SKV#4 dropped Index([], dtype='int64')
Sym Einstich SKV#4 dropped Index([], dtype='int64')
Sym aussen SKV#4 dropped Index([], dtype='int64')
Licht-Einstich frei SKV#4 dropped Index([], dtype='int64')
Beacon INT0028 - BME280_RH dropped Index([], dtype='int64')
Beacon INT0028 - BMP280_PRESS dropped Index([], dtype='int64')
Beacon INT0028 - BMP280_TEMP dropped Index([], dtype='int64')

```

```
[10]: X.isna().sum()
```

```
[10]: timestamp_skv          0
      timediff_skv          7
      timestamp_sensor      0
      timediff_sensor       7
      SynchroDiff           0
      ..
      Beacon INT0028 - BMP280_PRESS  0
      timestamp.2            0
      Beacon INT0028 - BMP280_TEMP  0
      pieces                 0
      tool io                 0
      Length: 165, dtype: int64
```

```
[13]: len(interestingColumns)
```

```
[13]: 147
```

### 1.1 Let's reduce the number of features; do a PCA

first we need to scale our features

e.g:  $(\text{value} - \text{min\_value}) / (\text{max\_value} - \text{min\_value})$

```
[15]: from sklearn.preprocessing import MinMaxScaler

      scaler = MinMaxScaler()

      scaler.fit(X[interestingColumns])

      Xscaled = pd.DataFrame(scaler.
      ↪transform(X[interestingColumns]), columns=interestingColumns)
```

```
[ ]: for f in interestingColumns:
      plt.plot(Xscaled.index, Xscaled[f])
      plt.title(f)
      plt.show()
```

```
[18]: from sklearn import decomposition

      pca = decomposition.PCA(n_components=len(interestingColumns))

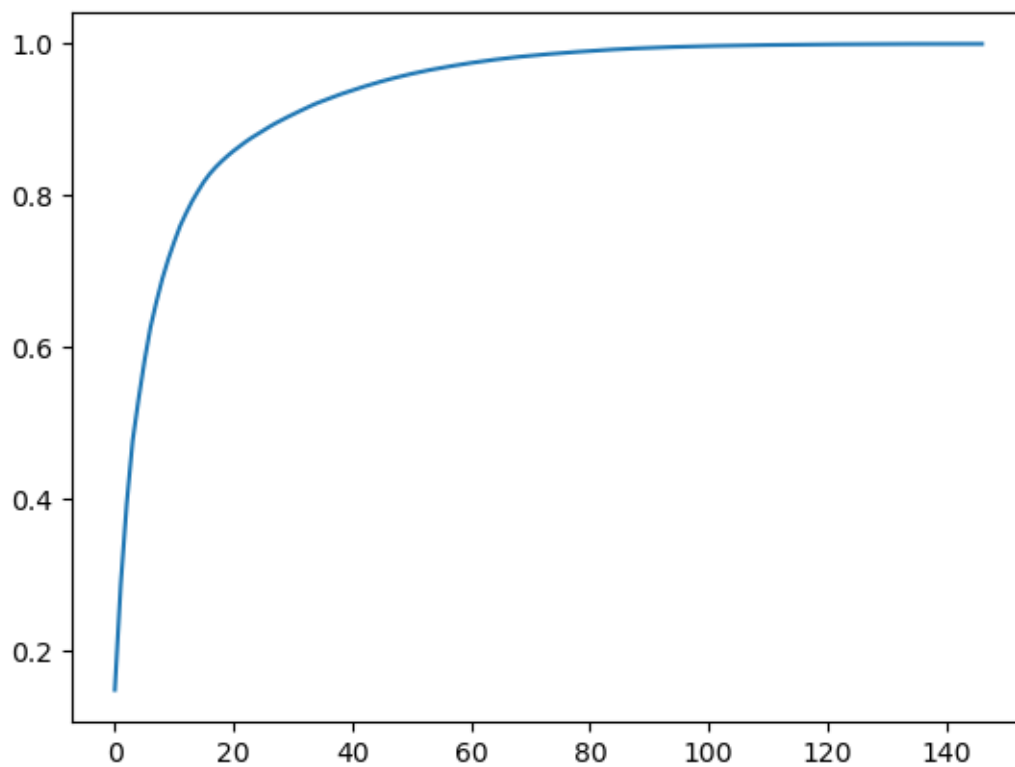
      pca.fit(Xscaled)

      Xpca = pd.DataFrame(pca.transform(Xscaled))
```

```
[22]: plt.plot(np.cumsum(pca.explained_variance_ratio_))
```

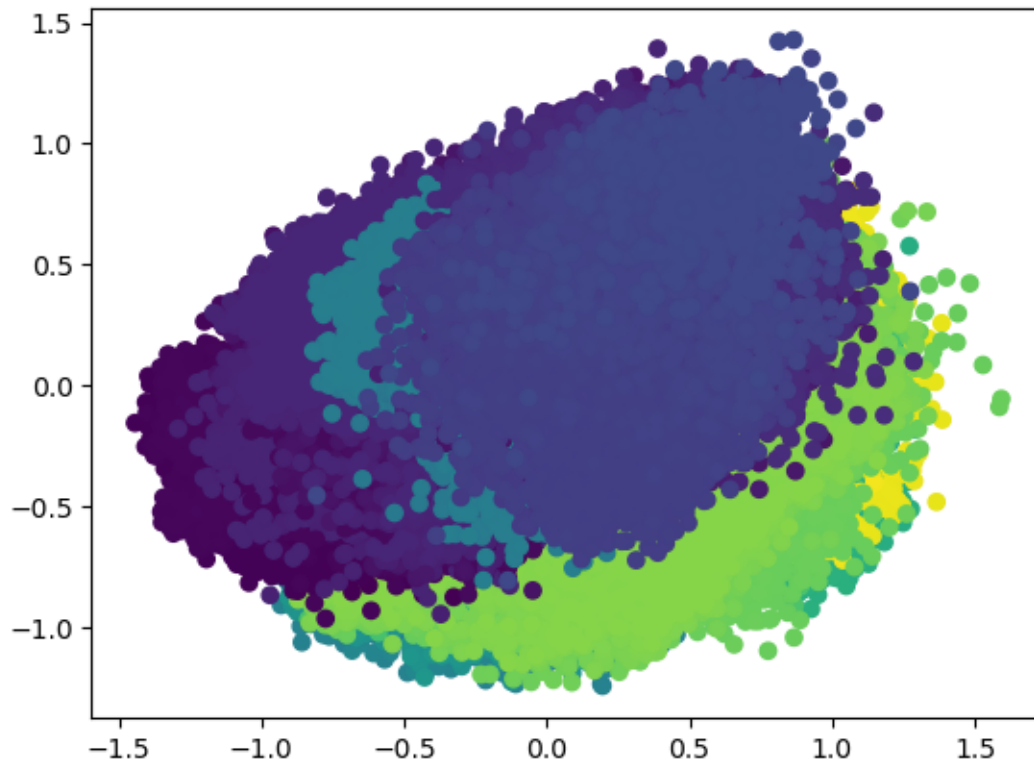
```
[22]: [<matplotlib.lines.Line2D at 0x7f871b4aff70>]
```





```
[26]: plt.scatter(Xpca[0],Xpca[1],c=X.pieces)
```

```
[26]: <matplotlib.collections.PathCollection at 0x7f871ba62f50>
```



```
[39]: movingAverageWindow = 500

Xavg = Xscaled.copy()

for col in Xscaled.columns:
    Xavg.loc[:,col] = Xscaled[col].rolling(movingAverageWindow).mean()

Xavg = Xavg.bfill(axis=0)

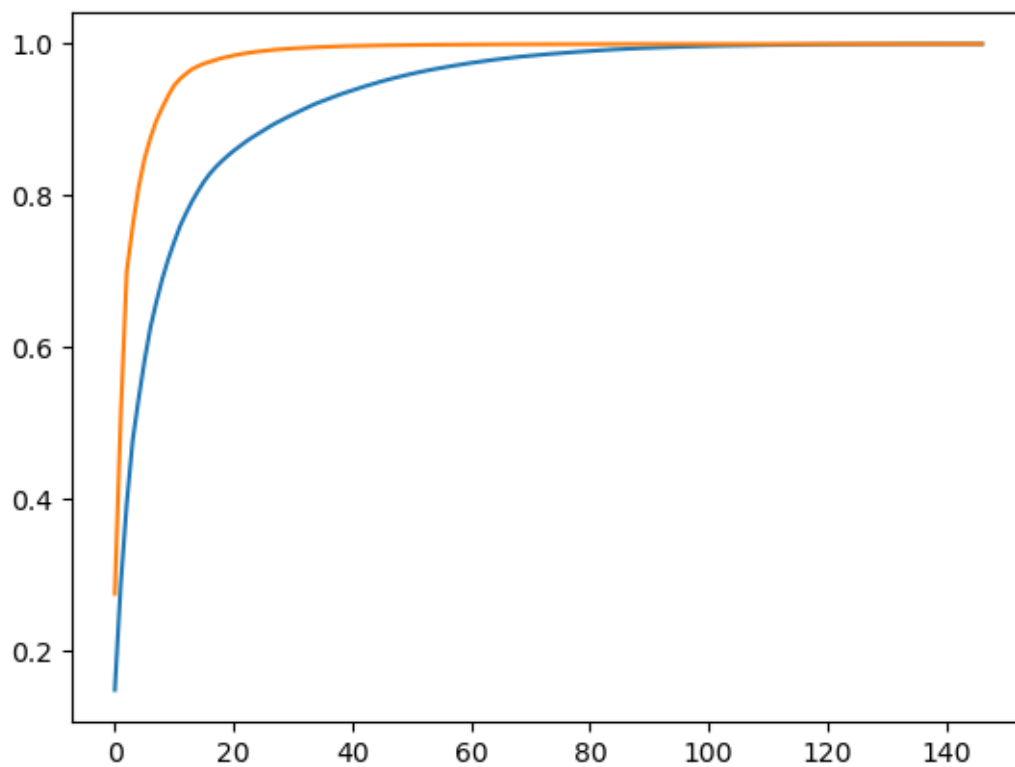
pcaAvg = decomposition.PCA(n_components=len(interestingColumns))

pcaAvg.fit(Xavg)

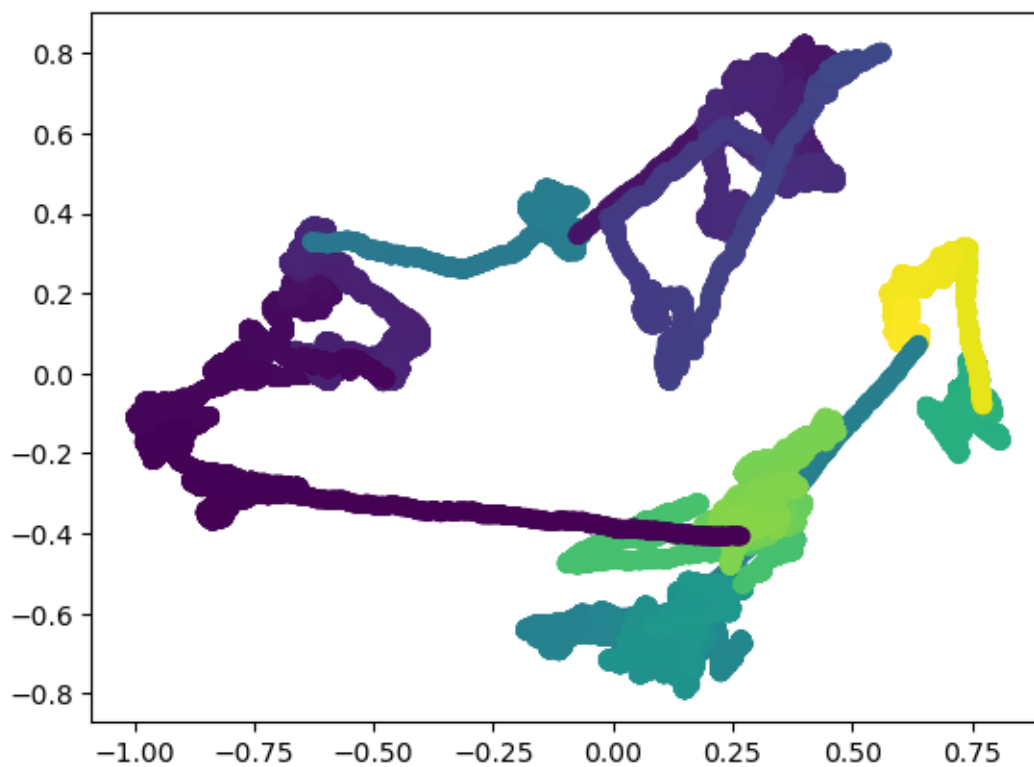
XpcaAvg = pd.DataFrame(pca.transform(Xavg))

plt.plot(np.cumsum(pca.explained_variance_ratio_))
plt.plot(np.cumsum(pcaAvg.explained_variance_ratio_))
plt.show()

plt.scatter(XpcaAvg[0],XpcaAvg[1],c=X.pieces)
```



[39]: <matplotlib.collections.PathCollection at 0x7f8721030670>



[ ]: