Data-Preparation

August 1, 2024

0.0.1 Needed Libraries for loading and manipulating csv files

[ ]:

```
import pandas as pd
```

[ ]:

```
filelocation = r'data/2017-09-01_EPS_BAT_TEMPS.csv'
```

[ ]:

```
df = pd.read_csv(filelocation, parse_dates=["created_on"])
df['temperature'] = df['temperature'].apply(lambda x: str(x.replace(',', '.')))
df['temperature'] = pd.to_numeric(df['temperature'])
```

[ ]:

```
newDF = pd.DataFrame()
newDF = newDF.join(df[df.sensor_id == 27]['created_on'])
newDF['created_on'] = df[df.sensor_id == 27]['created_on'].values
newDF['sensor_27'] = df[df.sensor_id == 27]['temperature'].values
newDF['sensor_28'] = df[df.sensor_id == 28]['temperature'].values
newDF['sensor_29'] = df[df.sensor_id == 29]['temperature'].values
newDF['is_anomaly'] = 0
newDF = newDF.rename(columns={"created_on": "timestamp"})

newDF = newDF.set_index('timestamp').shift(periods=2, freq="h")
newDF = newDF.reset_index()
```

0.0.2 Annotate the reported anomaly occurance dates

[ ]:

```
# 28/9/2017  13:50 to 28/9/2017 19:00
mark = (newDF['timestamp'] >
        '2017-09-28 13:50:00') & (newDF['timestamp'] <= '2017-09-28 19:00:00')
newDF.loc[mark, 'is_anomaly'] = 1
```

[ ]:

```
# 20/09/17 17:05 - 17:11
mark = (newDF['timestamp'] >
        '2017-09-20 17:05:00') & (newDF['timestamp'] <= '2017-09-20 17:11:00')
newDF.loc[mark, 'is_anomaly'] = 1
```

[ ]:

```
# 21/09/17 18:21 - 18:25
mark = (newDF['timestamp'] >
        '2017-09-21 18:21:00') & (newDF['timestamp'] <= '2017-09-21 18:25:00')
```

```
newDF.loc[mark, 'is_anomaly'] = 1
```

[ ]:
```
# 22/09/17 21:59 - 23:00
mark = (newDF['timestamp'] >
        '2017-09-22 21:59:00') & (newDF['timestamp'] <= '2017-09-22 23:00:00')
newDF.loc[mark, 'is_anomaly'] = 1
```

[ ]:
```
# 02/09/2017 5:12 - 04/09/2017 10:56
mark = (newDF['timestamp'] >
        '2017-09-02 05:12:00') & (newDF['timestamp'] <= '2017-09-04 10:56:00')
newDF.loc[mark, 'is_anomaly'] = 1
```

[ ]:
```
# 05/09/2017 15:00 - 20:28
mark = (newDF['timestamp'] >
        '2017-09-05 15:00:00') & (newDF['timestamp'] <= '2017-09-05 20:28:00')
newDF.loc[mark, 'is_anomaly'] = 1
```

[ ]:
```
# 06/09/2017 6:30 - 11:16
mark = (newDF['timestamp'] >
        '2017-09-06 06:30:00') & (newDF['timestamp'] <= '2017-09-06 11:16:00')
newDF.loc[mark, 'is_anomaly'] = 1
```

[ ]:
```
newDF.to_csv(r'data/MOVE_II_EPS_BAT_TEMPS_FULL_DATA.csv', index=False)
```

0.0.3 Create a train and test data for the semi-supervise algorithms

[ ]:
```
trainData = newDF.loc[newDF['is_anomaly'] == 0]
testData = newDF.loc[newDF['is_anomaly'] == 1]
```

[ ]:
```
n = 0.2
someGoodData = trainData.tail(int(trainData.shape[0]*n))
fulltestdf = pd.merge_ordered(testData, someGoodData)
fulltestdf.head(5)
```

[ ]:
```
trainData.to_csv(r'data/MOVE_II_EPS_BAT_TEMPS_TRAIN_DATA.csv', index=False)
fulltestdf.to_csv(r'data/MOVE_II_EPS_BAT_TEMPS_TEST_DATA.csv', index=False)
```