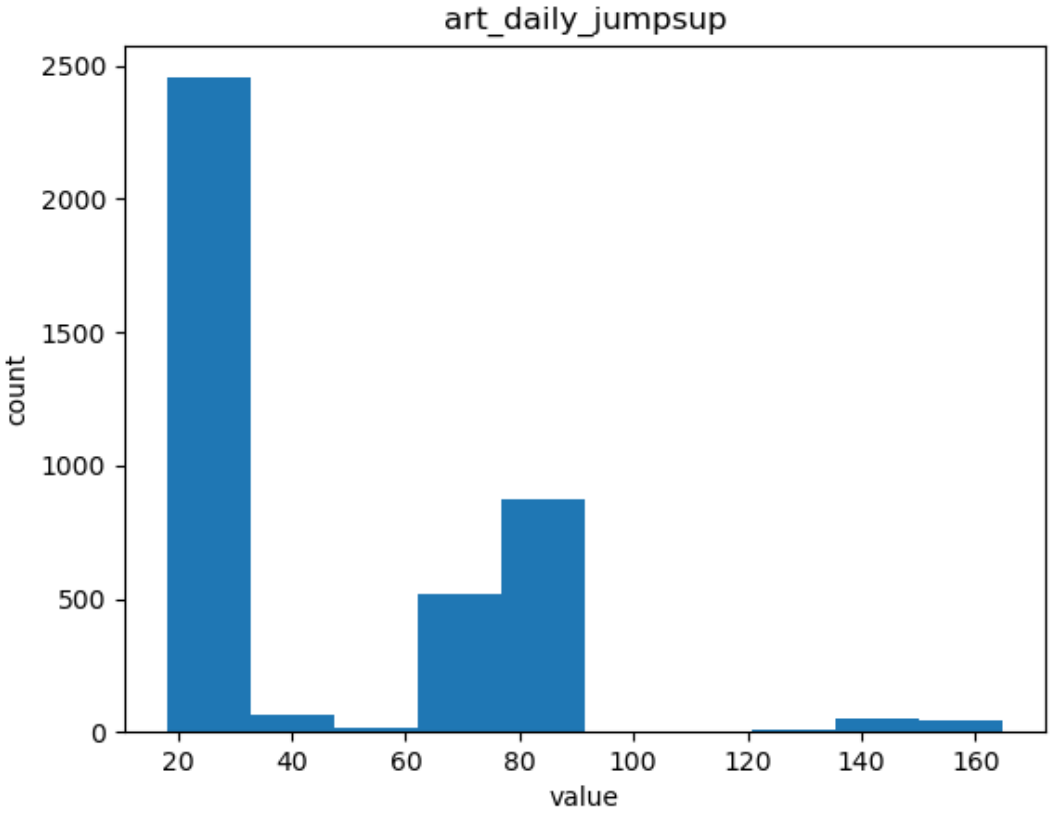
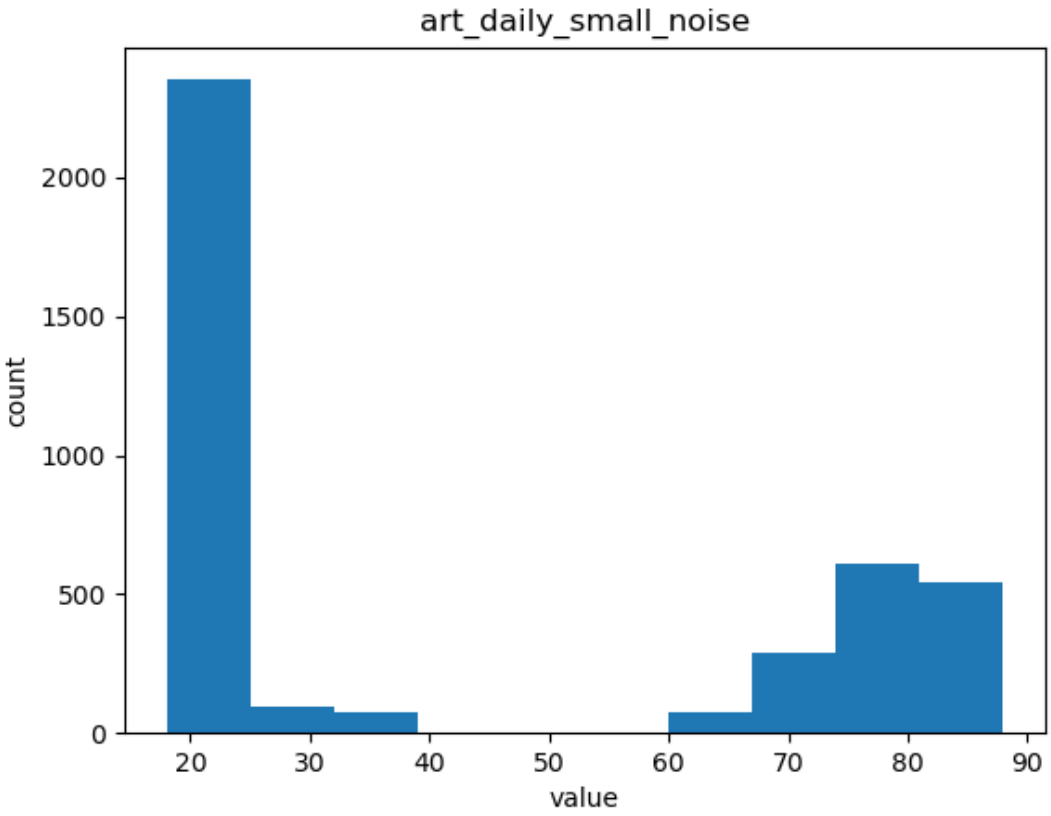
Timeseries anomaly detection using an Autoencoder

This script demonstrates how you can use a reconstruction convolutional autoencoder model to detect anomalies in timeseries data.

1.Load the data

The data set is loaded first, the training file is art_daily_small_noise.csv, and the test file is art_daily_jumpsup.csv.The distribution of data is shown in the following figure:





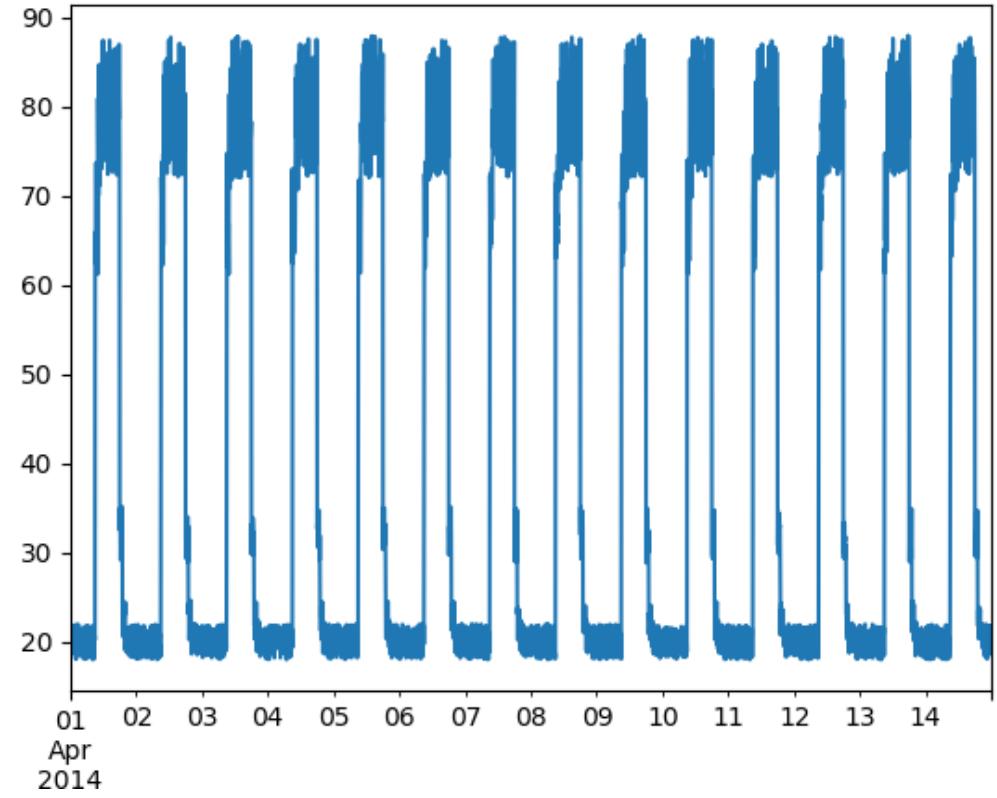
Checking to see if there is an NAn value:

art_daily_small_noise doesn't have Nan value.

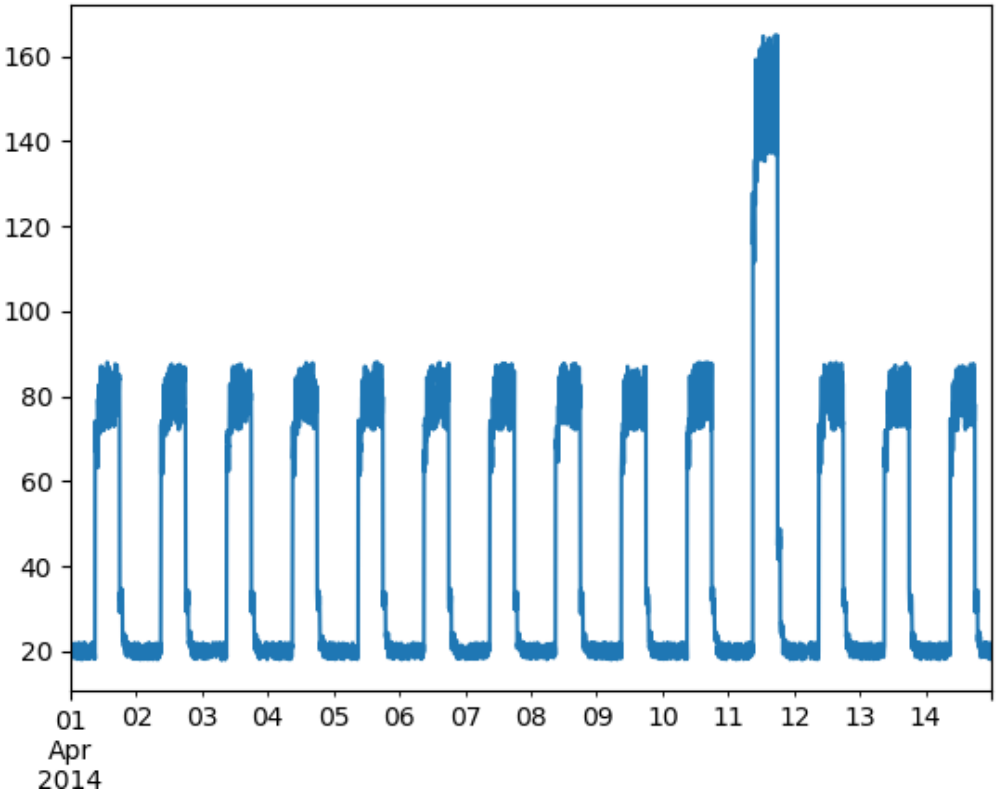
art_daily_jumpsup doesn't have Nan value.

Display data in a time series,

Timeseries data without anomalies



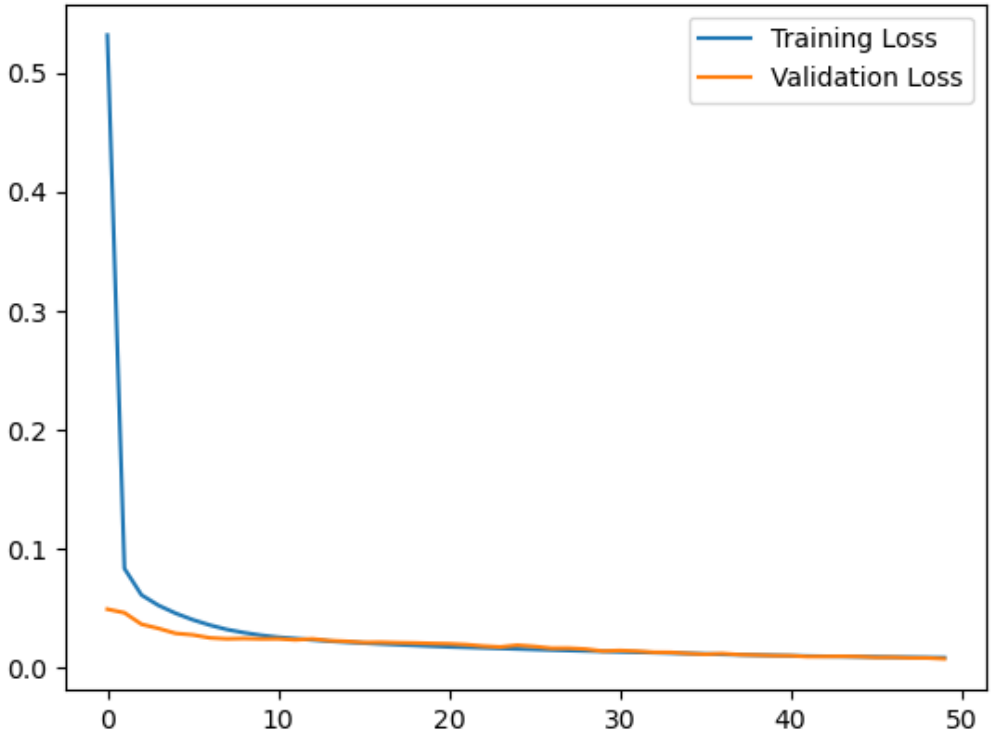
Timeseries data with anomalies



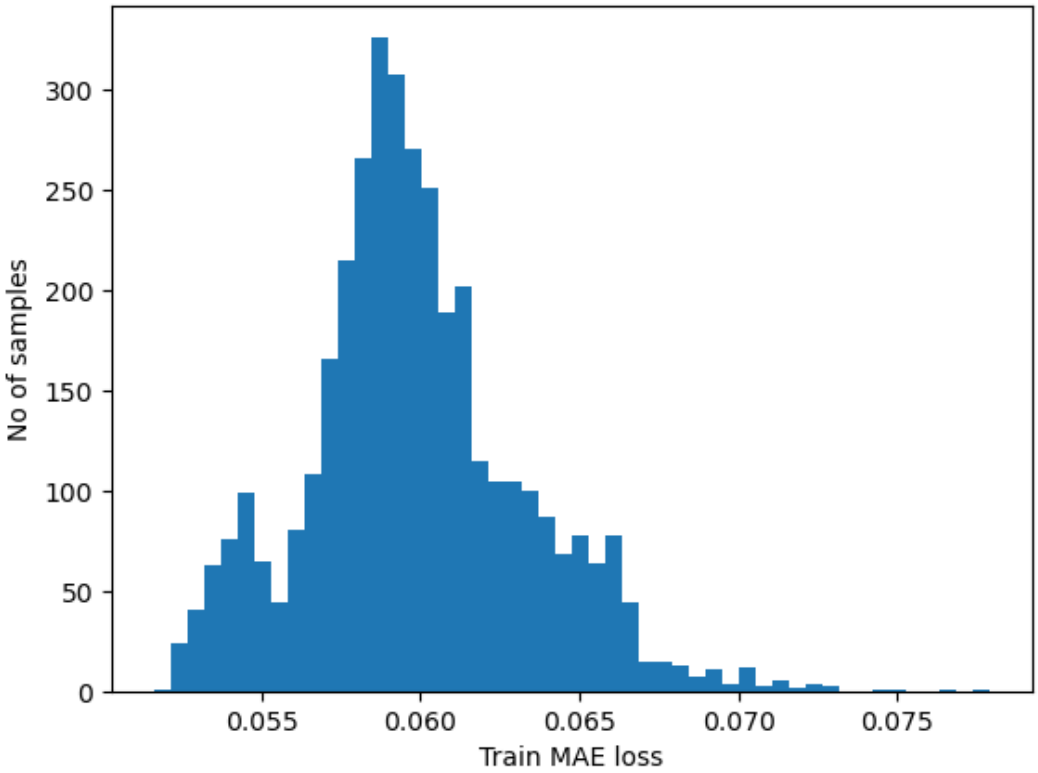
Build a model and Train the model

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d (Conv1D)              (None, 144, 32)           256
_________________________________________________________________
dropout (Dropout)            (None, 144, 32)           0
_________________________________________________________________
conv1d_1 (Conv1D)            (None, 72, 16)            3600
_________________________________________________________________
conv1d_transpose (Conv1DTran (None, 144, 16)           1808
_________________________________________________________________
dropout_1 (Dropout)          (None, 144, 16)           0
_________________________________________________________________
conv1d_transpose_1 (Conv1DTr (None, 288, 32)           3616
_________________________________________________________________
conv1d_transpose_2 (Conv1DTr (None, 288, 1)            225
=================================================================
Total params: 9,505
Trainable params: 9,505
Non-trainable params: 0
_________________________________________________________________
Epoch 1/50
27/27 [==============================] - 1s 39ms/step - loss: 0.5906 - val_loss: 0.3041
Epoch 2/50
27/27 [==============================] - 1s 33ms/step - loss: 0.1232 - val_loss: 0.0525
Epoch 3/50
27/27 [==============================] - 7s 247ms/step - loss: 0.0570 - val_loss: 0.0374
Epoch 4/50
27/27 [==============================] - 1s 32ms/step - loss: 0.0456 - val_loss: 0.0302
```

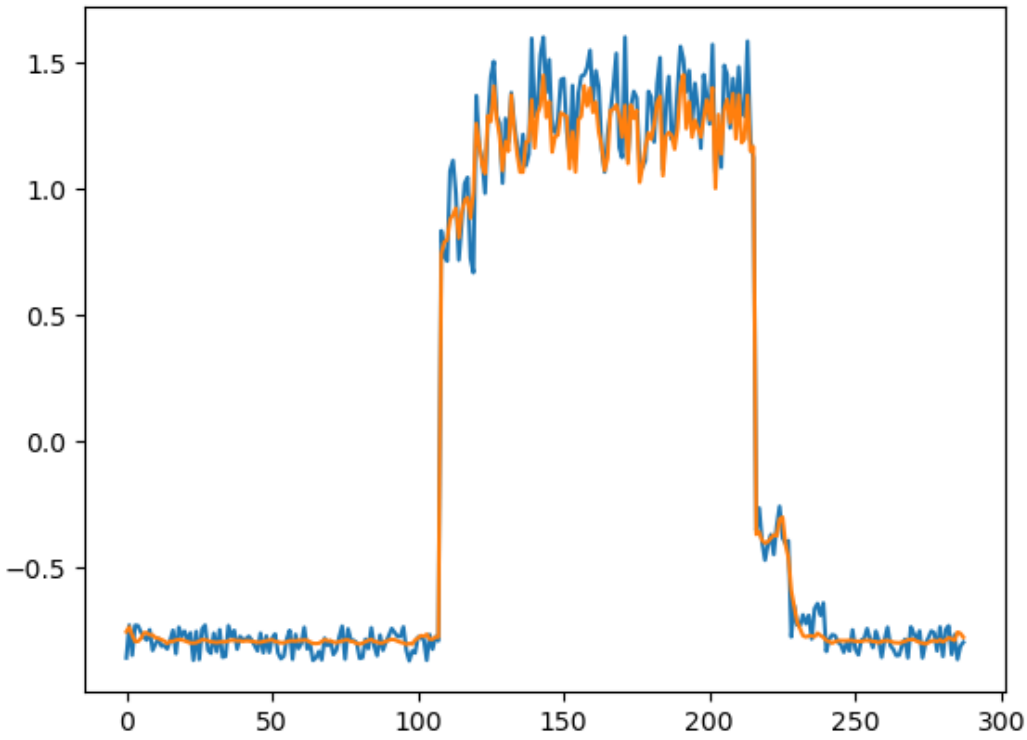
plot training and validation loss



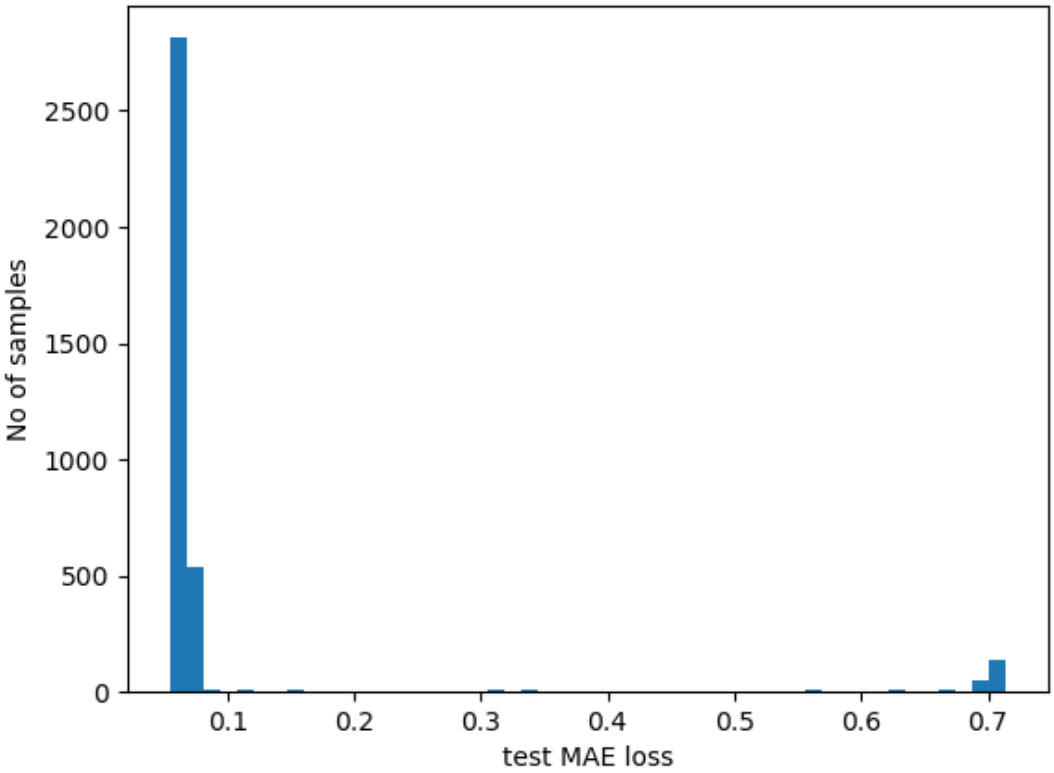
Mean Absolute Error



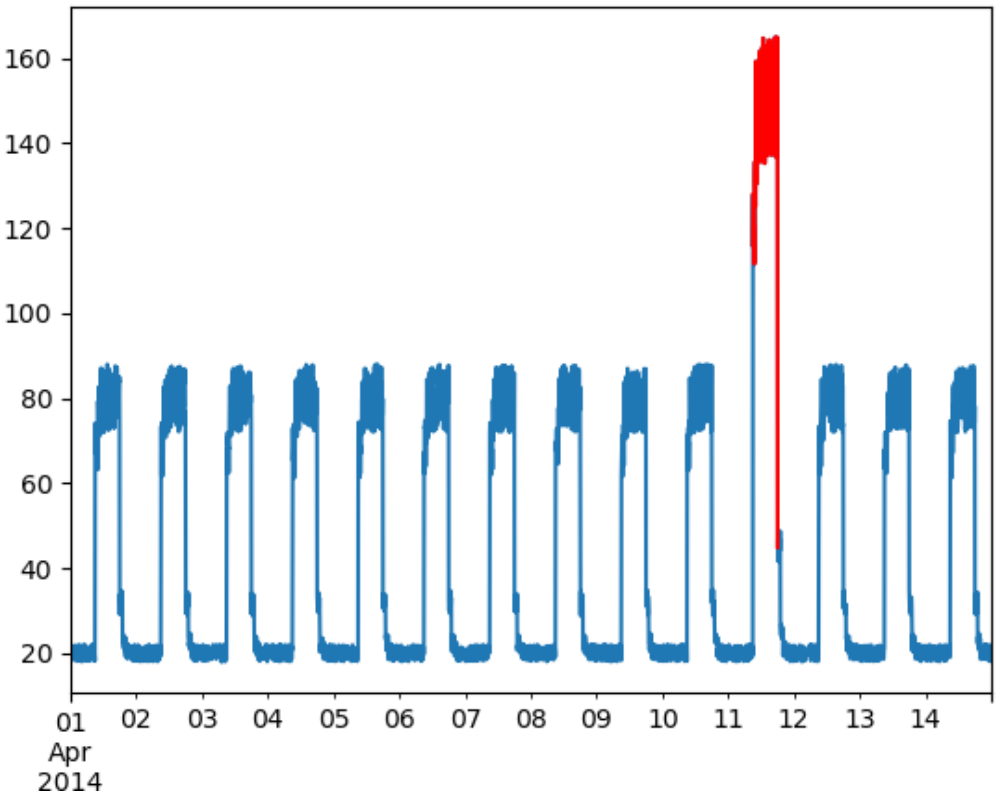
Compare recontruction



Test MEA loss



overlay the anomalies on the original test data plot.



Conclusion:

This script will detect anomalies by finding MAE loss. The max MAE loss value will be the 'threshold' for anomaly detection. Finally, the samples of the data which are anomalies are highlighted in the figure. The advantage of this approach is that you don't have to set the threshold manually. But there is a slight lag in time.