

Cad_Phase-4

Development of Serverless IOT Data Processing: [part-2]

Modeling: What are we trying to solve?

Case 1: Change Detection: Detecting excessive energy consumption in advance and preventing increase in usage fees.

Case 2: Predict Future Consumption: Predicting future energy consumption and generation by utilizing weather information and optimizing energy supply.

Case 1: Change detection

The change point is the point at which the trends in time series data change over time. Outliers indicate a momentary abnormal condition (rapid decrease or increase), while change points mean that the abnormal condition does not return to its original state and continue.

Let's use Change Finder algorithm

Change Finder is an algorithm used to detect change points. Change Finder uses the log-likelihood based on the SDAR (Sequencially Discounting AR) algorithm to calculate the change score. SDAR algorithm introduces a discounting parameter into the AR algorithm to reduce the influence of past data, so that even non-stationary time series data can be learned robustly.

Change Finder has two steps of model training:

Training STEP1 Train a time series model at each data point using the SDAR algorithm Based on the trained time series model, calculate the likelihood that the data points at the next time point will appear Calculate the logarithmic loss and use it as an outlier score

$$Score(x_t) = -\log P_{t-1}(x_t | x_1, x_2, \dots, x_{t-1})$$

Smoothing Step Smooth the outlier score within the smoothing window(W).
By smoothing, the score due to outliers is attenuated, and it is possible to determine whether the abnormal condition has continued for a long time.

$$Score_smoothed(xt)=\frac{1}{W}\sum_{t=t-W+1}^t Score(xi)$$

Training STEP2 Using the score obtained by smoothing, train the model with the SDAR algorithm Based on the trained time series model, calculate the likelihood that the data points at the next time point will appear Calculate the logarithmic loss and use it as an change score

In [1]:

```
# Let's install & import changefinder python library
```

```
!pip install changefinder
```

```
import changefinder
```

OUT [1]:

Defaulting to user installation because normal site-packages is not writeable

Requirement already satisfied: changefinder in ./local/lib/python3.9/site-packages (0.3)

Requirement already satisfied: scipy in /opt/tljh/user/lib/python3.9/site-packages (from changefinder) (1.5.4)

Requirement already satisfied: nose in ./local/lib/python3.9/site-packages (from changefinder) (1.3.7)

Requirement already satisfied: numpy in /opt/tljh/user/lib/python3.9/site-packages (from changefinder) (1.20.3)

Requirement already satisfied: statsmodels in /opt/tljh/user/lib/python3.9/site-packages (from changefinder) (0.13.2)

Requirement already satisfied: pandas>=0.25 in /opt/tljh/user/lib/python3.9/site-packages (from statsmodels->changefinder) (1.5.1)

Requirement already satisfied: patsy>=0.5.2 in /opt/tljh/user/lib/python3.9/site-packages (from statsmodels->changefinder) (0.5.3)

Requirement already satisfied: packaging>=21.3 in /opt/tljh/user/lib/python3.9/site-packages (from statsmodels->changefinder) (21.3)

Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /opt/tljh/user/lib/python3.9/site-packages (from packaging>=21.3->statsmodels->changefinder) (3.0.9)

Requirement already satisfied: python-dateutil>=2.8.1 in /opt/tljh/user/lib/python3.9/site-packages (from pandas>=0.25->statsmodels->changefinder) (2.8.2)

Requirement already satisfied: pytz>=2020.1 in /opt/tljh/user/lib/python3.9/site-packages (from pandas>=0.25->statsmodels->changefinder) (2022.6)

Requirement already satisfied: six in /opt/tljh/user/lib/python3.9/site-packages (from patsy>=0.5.2->statsmodels->changefinder) (1.16.0)

In [22]:

```
from scipy import stats
import holoviews as hv
from holoviews import opts
hv.extension('bokeh')
```

OUT [22]:

```
cf = changefinder.ChangeFinder(r=_r, order=_order, smooth=_smooth)
```

```
ch_df = pd.DataFrame()
```

```
ch_df[col] = df[col].resample('D').mean()
```

```
# calculate the change score
```

```
ch_df['change_score'] = [cf.update(i) for i in ch_df[col]]
```

```

ch_score_q1 = stats.scoreatpercentile(ch_df['change_score'], 25)
ch_score_q3 = stats.scoreatpercentile(ch_df['change_score'], 75)
thr_upper = ch_score_q3 + (ch_score_q3 - ch_score_q1) * 3

anom_score = hv.Curve(ch_df['change_score'])
anom_score_th = hv.HLine(thr_upper).opts(color='red', line_dash="dotdash")

anom_points = [[ch_df.index[i],ch_df[col][i]] for i, score in
enumerate(ch_df["change_score"]) if score > thr_upper]

org = hv.Curve(ch_df[col],label=col).opts(yformatter='%.1fkW')
detected = hv.Points(anom_points, label=f'{col} detected').opts(color='red',
legend_position='bottom', size=5)

return ((anom_score * anom_score_th).opts(title=f'{col} Change Score &
Threshold') + \
        (org * detected).opts(title=f'{col} Detected
Points')).opts(opts.Curve(width=800, height=300, show_grid=True,
tools=['hover'])).cols(1)

```

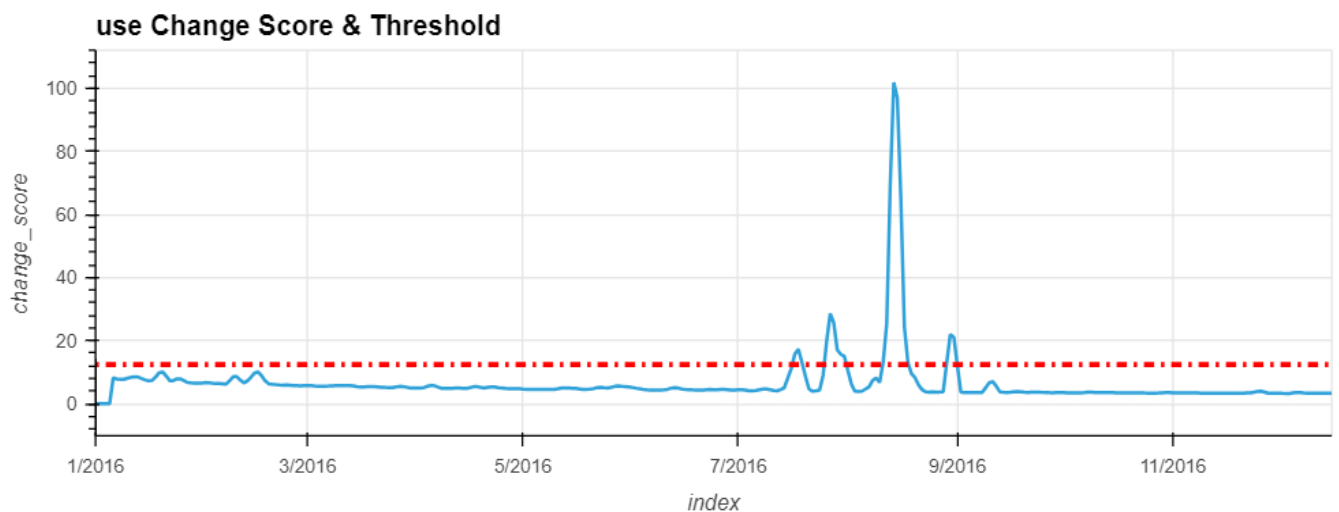
Discounting parameter $r(0 < r < 1)$: The smaller this value, the greater the influence of the past data points and the greater the variation in the change score

Order parameter for AR *order*: How far past data points are included in the model

Smoothing window *smooth*: The greater this parameter is, the easier it is to capture the essential changes rather than the outliers, but if it is too large, it will be difficult to capture the changes themselves.

```
In [4]: chng_detection('use', _r=0.001, _order=1, _smooth=3)
```

OUT [4]:



```
In [5]: df.columns
```

OUT [5]:

```
Index(['use', 'gen', 'Dishwasher', 'Furnace_1', 'Furnace_2',  
      'Home_office', 'Fridge', 'Wine_cellar', 'Garage_door', 'Kitchen_12',  
      'Kitchen_14', 'Kitchen_38', 'Barn', 'Well', 'Microwave', 'Living_room',  
      'temperature', 'icon', 'humidity', 'visibility', 'summary',  
      'apparentTemperature', 'pressure', 'windSpeed', 'cloud Cover',
```

```
'windBearing', 'precipIntensity', 'dewPoint', 'precipProbability',  
'kitchen', 'Furnace'], dtype='object')
```

```
In [25]: Chng detection('Furnace', _r=0.001, _order=1, _smooth=3)
```

OUT [25]:

