ADS: Rapid Deployment of Anomaly Detection Models

Jiahao Bu

Tsinghua university

Outline

- Background
- Problem definition
- Design
- Evaluation

Outline

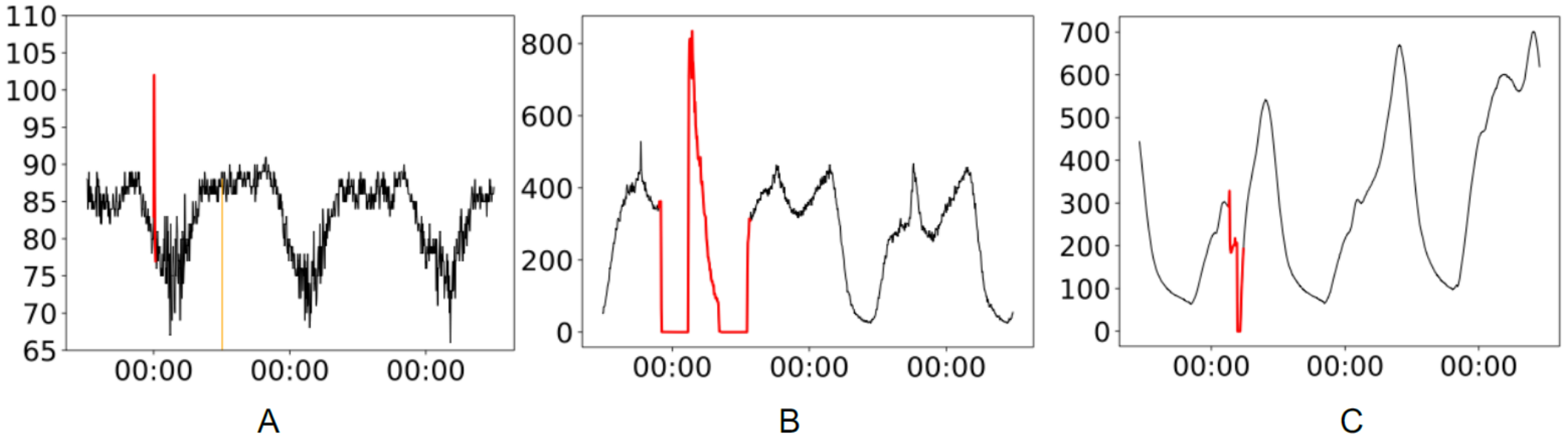
- Background ⬅
- Problem definition
- Design
- Evaluation

Background

- Internet-based services (e.g., online games, online shopping, social networks, search engine) monitor KPIs (Key Performance Indicators) of their applications and systems in order to keep their services reliable.
  - E.g., CPU utilization, number of queries per second, response latency

- Anomalies on KPI likely indicate underlying failures on Internet services
  - E.g., a spike or dip in a KPI stream

Background



Examples of anomalies in KPI streams. The red parts in the KPI stream denote anomalous points, and the orange part denotes missing points (filled with zeros).

Background

However, there remains one common and important scenario that large number of KPI streams emerge *continuously and frequently,* which has not been studied !!!!

Background

Case 1:

- New products can be frequently launched, such as in gaming platform. For example, in a top gaming company G studied in this paper, on average over ten new games are launched per quarter, which results in more than 6000 new KPI streams per 10 days on average.

Background

Case 2:

- With the popularity of DevOps and micro-service, software upgrades become more and more frequent, many of which result in the pattern changes of existing KPI streams, making the previous anomaly detection algorithms/parameters outdated.

Outline

- Background
- Problem definition ⬅
- Design
- Evaluation

Problem definition

In the above scenario, the algorithm needs to overcome the following difficulties while maintaining high performance:

- manual algorithm selection
- parameter tuning
- new anomaly labeling

Problem definition

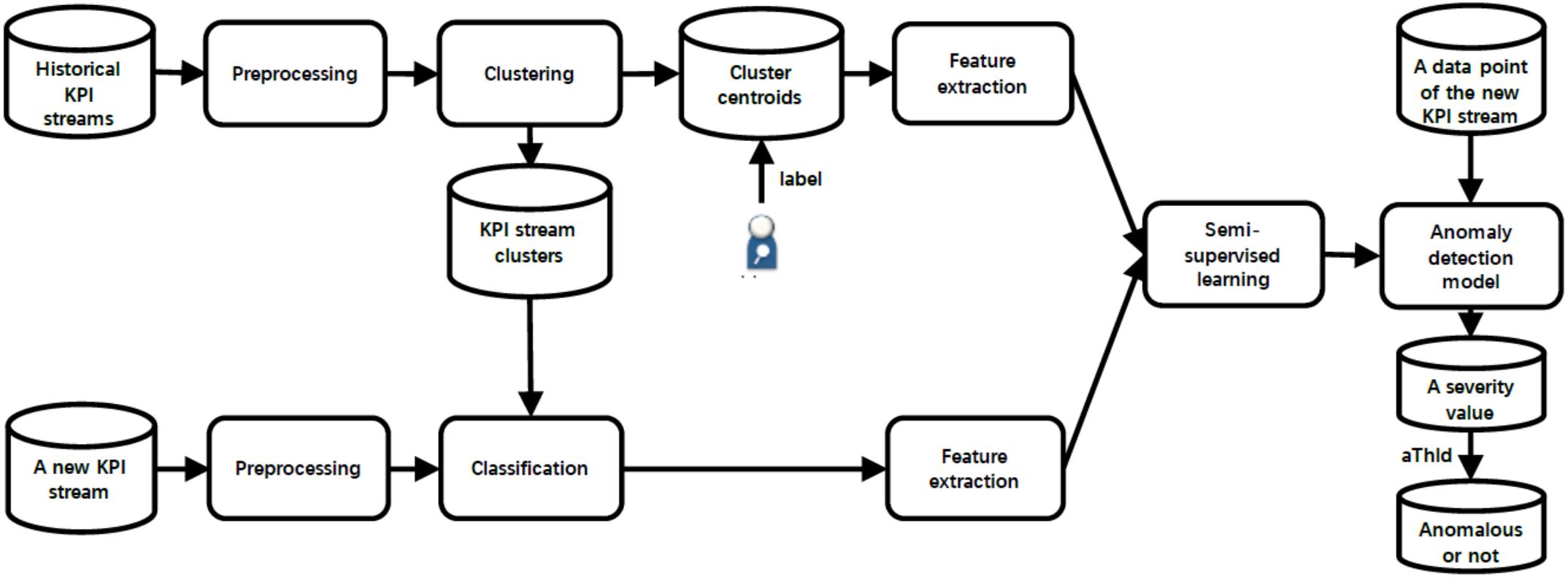
Unfortunately, none of the existing anomaly detection approaches are feasible to deal with the above scenario well

- **Traditional statistical algorithms** often need manual algorithm selection parameter tuning
- **Supervised learning based methods** require manually labeling anomalies for each new KPI stream
- **Unsupervised learning based methods** suffer from low accuracy or require large amounts of training data for each new KPI stream

Outline

- Background
- Problem definition
- Design ⬅
- Evaluation

Design

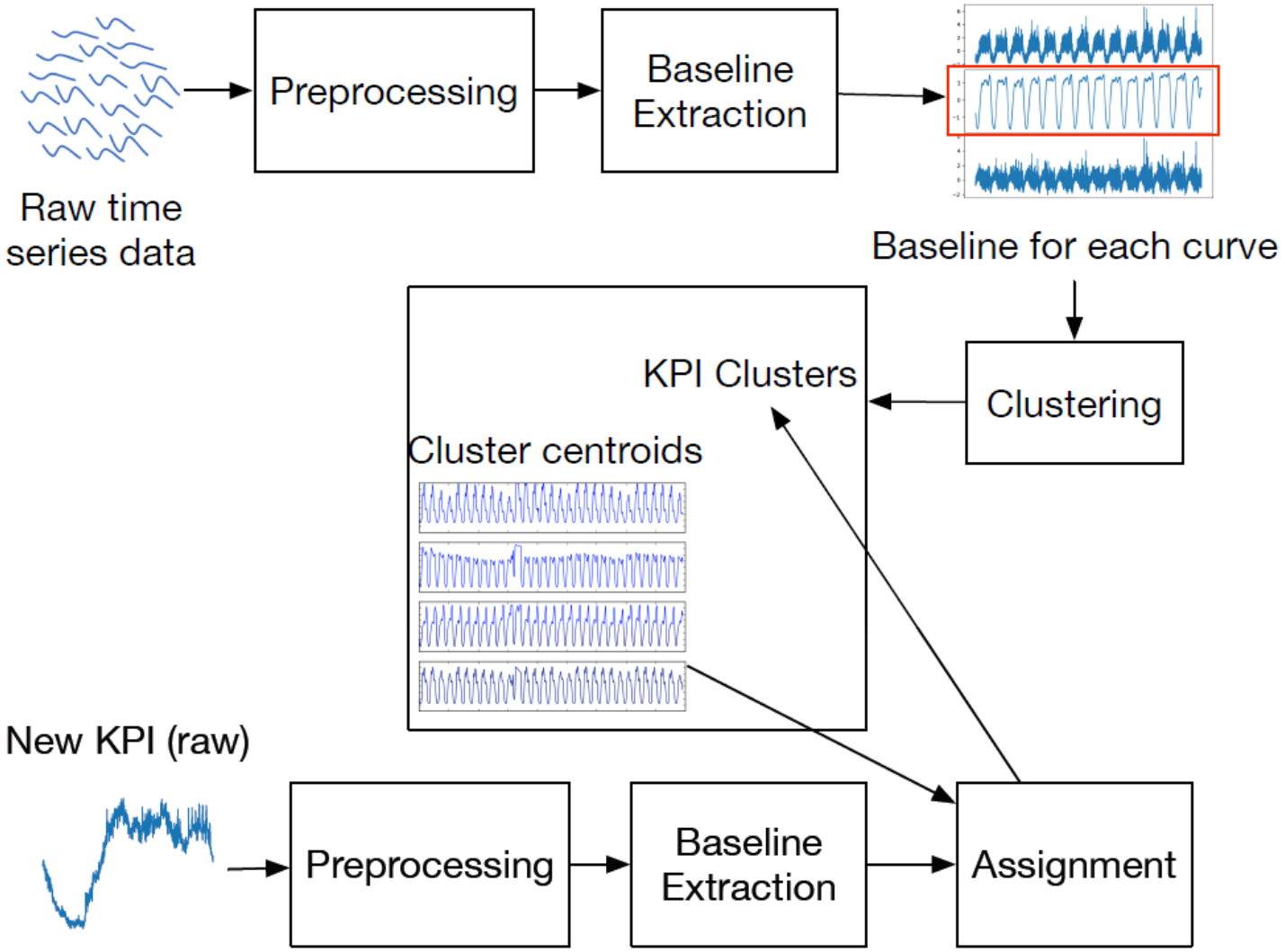


ADS proposes to cluster all existing/historical KPI streams into clusters, assign each newly emerging KPI stream into one of the existing clusters, and then combine the data of the new KPI stream (unlabeled) and it's cluster centroid (labeled) and use semi-supervised learning to train a new model for each new KPI stream.

Preprocessing

- Fill these missing points using linear interpolation
- Standardization

Clusering



- ADS adopts ROCKA to group KPI streams into a few clusters.
- Then we obtain a centroid KPI stream for each cluster and can label anomaly points.

Feature extraction

Detectors / #Configurations	Sampled Parameters
Simple threshold [25] / 1	none
Diff / 3	last-point, last-day, last-week
Simple MA [26] / 5	win = 10, 20, 30, 40, 50 points
Weighted MA [27] / 5	
MA of diff / 5	
EWMA [27] / 5	α = 0.1, 0.3, 0.5, 0.7, 0.9
TSD [1] / 1	win = 1 week
TSD MAD / 1	
Historical average [24] / 1	
Historical MAD / 1	
Holt-Winters [9] / $4^3 = 64$	α, β, γ = 0.2, 0.4, 0.6, 0.8
SVD [6] / $5 \times 3 = 15$	#row =10, 20, 30, 40, 50 points, #column =3, 5, 7
Wavelet [7] / $3 \times 3 = 9$	win = 3, 5 ,7 days, freq = low, mid, high
ARIMA [8] / 1	Estimation from data
In total: 14 detectors / 117 configurations	

Feature: Difference value of predict KPI and actual KPI.

Detector: Predict algorithm with a certain parameter.

Feature vector: All feature values extracted by a specific detector and sorted by time.

Semi-Supervised Learning

In this work, we adopt CPLE , an extension model of self-training.

CPLE has the four following advantages:

- CPLE is flexible to change base-model
- CPLE needs low memory complexity
- CPLE is more robust than other semi-supervised learning algorithms
- CPLE supports incremental learning.

Semi-Supervised Learning

In addition, the negative log loss for binary classifiers takes on the general form:

$$J(\mathbf{y}, \mathbf{p}) = \log \mathbf{p}(\mathbf{y}|\mathbf{p})$$
$$= \frac{1}{N} \sum_{i=1}^{N} [y_i \log p_i + (1 - y_i) \log(1 - p_i)]$$

where N is the number of the data points in the KPI streams of training set, y_i is the label of the i-th data point and p_i is the i-th discriminative likelihood (DL)

Semi-Supervised Learning

The objective of CPLE is to minimize the function:

$$E(\mathbf{q}, \theta | \mathbf{X}, \mathbf{U}) = J(\mathbf{y}', g(\mathbf{U}; \theta)) - J(\mathbf{y}, g(\mathbf{X}; \theta))$$

where X is the data set of labeled data points, U is the one of unlabeled data points, and y' = H(q), where:

$$H(q_i) = \begin{cases} 1 & \text{if } q_i \geq 0.5 \\ 0 & \text{if } q_i < 0.5 \end{cases}$$

This way, (the parameter vector of) the base-model, which serves as the anomaly detection model, is trained based on (X U U) using actual and hypothesized labels (y U y'), as well as the weights of data points w, where:

$$w_i = \begin{cases} 1 & \text{if } x_i \in \mathbf{X} \\ 0 & q_i \text{ otherwise} \end{cases}$$

Outline

- Background
- Problem definition
- Design
- Evaluation ⬅

Data Set

- We randomly pick 70 historical KPI streams for clustering and 81 new ones for anomaly detection from a top global online game service.

- The following table are description of 81 new ones :

KPI	# KPI streams	Interval (minute)	Length (month)	#/percentage of anomalous data points per KPI stream
Latency	19	5	1	150/1.7%
# online players	58	5	1	72/0.83%
Success rate	4	5	1	84/0.97%

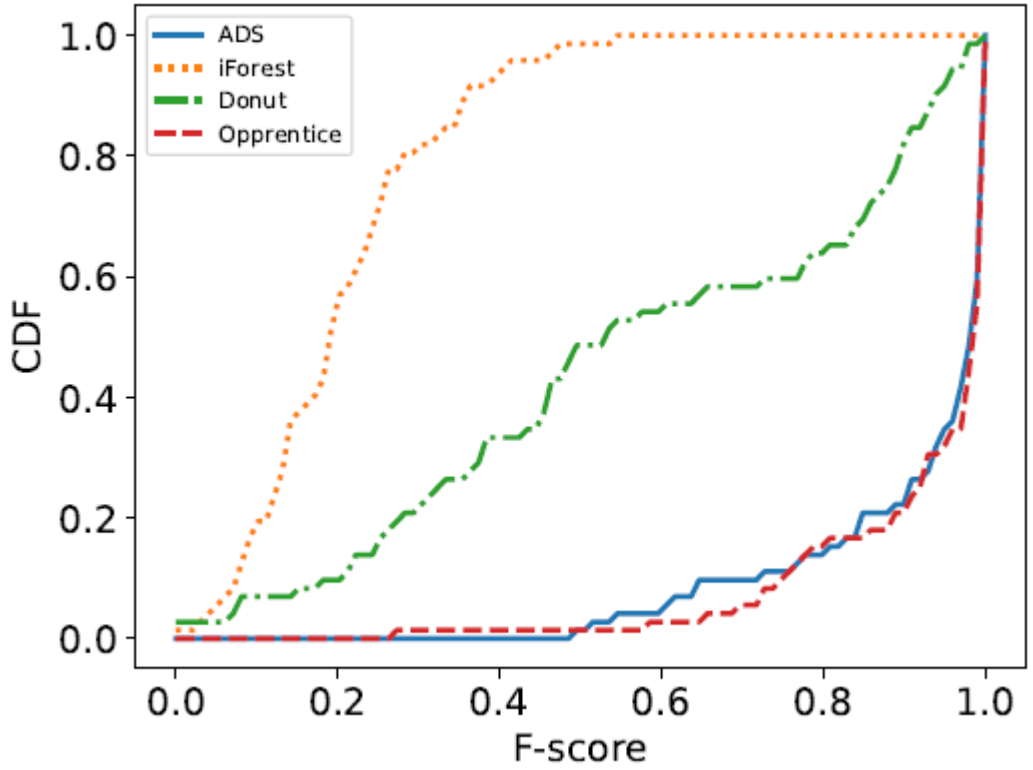
Evaluation of The Overall Performance

To evaluate the performance of ADS in anomaly detection for KPI streams, we calculate its best F-score, and compare it with that of iForest, Donut and Opprentice

Cluster	# KPI streams	*ADS*	iForest	Donut	Opprentice	ROCKA + Opprentice
A	7	0.91	0.33	0.42	0.90	0.67
B	9	0.91	0.21	0.37	0.91	0.88
C	8	0.95	0.22	0.28	0.98	0.94
D	53	0.93	0.19	0.67	0.94	0.90
E	4	0.67	0.13	0.45	0.71	0.66
Overall	81	0.92	0.20	0.57	0.93	0.87

Evaluation of The Overall Performance

CDFs of the best F-scores of each new KPI stream using ADS, iForest, Donut and Opprentice, respectively.



Evaluation of CPLE

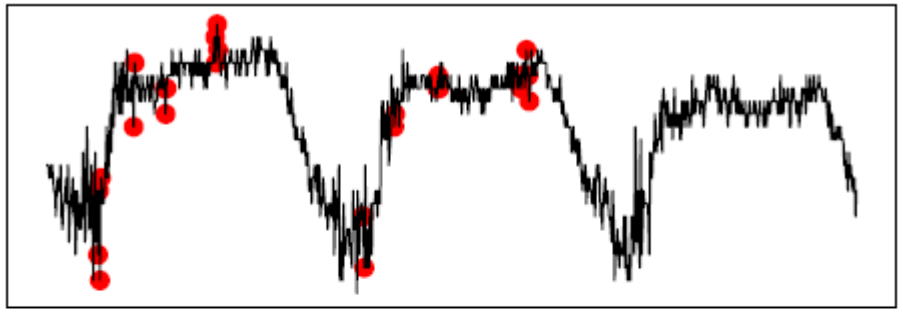
- To the best of our knowledge, this is the first work to apply semi-supervised learning CPLE to the KPI anomaly detection problem. We want to evaluate the performance of CPLE.

- The following table are new KPI streams where ADS performs significantly better than ROCKA + Opprentice.

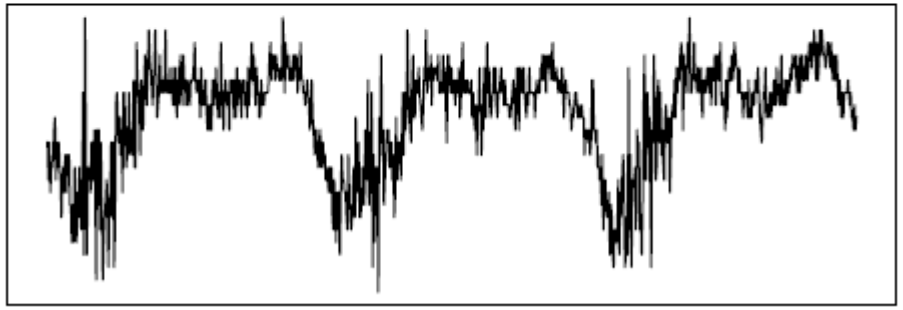
KPI stream ID	*ADS*	ROCKA + Opprentice
α	0.86	0.62
β	0.91	0.20
γ	0.72	0.46
δ	0.80	0.55
...	...	...

Evaluation of CPLE

KPI stream clustering methods such as ROCKA usually extract baselines (namely underlying shapes) from KPI streams and ignore fluctuations. However, the fluctuations of KPI streams can impact anomaly detection.



KPI stream α



The KPI stream on the centroid

- The anomaly detection results of ROCKA + Opprentice on KPI stream α, and α's cluster centroid KPI stream.
- The red data points are anomalous determined by ROCKA + Opprentice while in actual they are normal.

Evaluation of CPLE

ADS addresses the above problem effectively using semisupervised learning. In other words, it learns not only from the labels of the centroid KPI stream, but also from the fluctuation degree of the new KPI stream.

THANKS !