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
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).
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.
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result in the pattern changes of existing KPI streams, making the previous anomaly detection algorithms/parameters outdated.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
In the above scenario, the algorithm needs to overcome the following difficulties while maintaining high performance:

- manual algorithm selection
- parameter tuning
- new anomaly labeling
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
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 its 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
• 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

<table>
<thead>
<tr>
<th>Detectors / #Configurations</th>
<th>Sampled Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple threshold [25] / 1</td>
<td>none</td>
</tr>
<tr>
<td>Diff / 3</td>
<td>last-point, last-day, last-week</td>
</tr>
<tr>
<td>Simple MA [26] / 5</td>
<td></td>
</tr>
<tr>
<td>Weighted MA [27] / 5</td>
<td>win = 10, 20, 30, 40, 50 points</td>
</tr>
<tr>
<td>MA of diff / 5</td>
<td></td>
</tr>
<tr>
<td>EWMA [27] / 5</td>
<td>$\alpha = 0.1, 0.3, 0.5, 0.7, 0.9$</td>
</tr>
<tr>
<td>TSD [1] / 1</td>
<td></td>
</tr>
<tr>
<td>TSD MAD / 1</td>
<td>win = 1 week</td>
</tr>
<tr>
<td>Historical average [24] / 1</td>
<td></td>
</tr>
<tr>
<td>Historical MAD / 1</td>
<td></td>
</tr>
<tr>
<td>Holt-Winters [9] / $4^3 = 64$</td>
<td>$\alpha, \beta, \gamma = 0.2, 0.4, 0.6, 0.8$</td>
</tr>
<tr>
<td>SVD [6] / $5 \times 3 = 15$</td>
<td>#row =10, 20, 30, 40, 50 points, #column =3, 5, 7</td>
</tr>
<tr>
<td>Wavelet [7] / $3 \times 3 = 9$</td>
<td>win = 3, 5, 7 days, freq = low, mid, high</td>
</tr>
<tr>
<td>ARIMA [8] / 1</td>
<td>Estimation from data</td>
</tr>
</tbody>
</table>

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.
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:

\[ J(y, p) = \log p(y|p) \]

\[ = \frac{1}{N} \sum_{i=1}^{N} [y_i \log p_i + (1 - y_i) \log(1 - p_i)] \]

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:

\[ E(q, \theta | X, U) = J(y', g(U; \theta)) - J(y, g(X; \theta)) \]

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:

\[ w_i = \begin{cases} 
1 & \text{if } x_i \in X \\
0 & q_i \text{ otherwise} 
\end{cases} \]
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:

<table>
<thead>
<tr>
<th>KPI</th>
<th># KPI streams</th>
<th>Interval (minute)</th>
<th>Length (month)</th>
<th>#/percentage of anomalous data points per KPI stream</th>
</tr>
</thead>
<tbody>
<tr>
<td>Latency</td>
<td>19</td>
<td>5</td>
<td>1</td>
<td>150/1.7%</td>
</tr>
<tr>
<td># online players</td>
<td>58</td>
<td>5</td>
<td>1</td>
<td>72/0.83%</td>
</tr>
<tr>
<td>Success rate</td>
<td>4</td>
<td>5</td>
<td>1</td>
<td>84/0.97%</td>
</tr>
</tbody>
</table>
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 Apprentice.

<table>
<thead>
<tr>
<th>Cluster</th>
<th># KPI streams</th>
<th>ADS</th>
<th>iForest</th>
<th>Donut</th>
<th>Apprentice</th>
<th>ROCKA + Apprentice</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>7</td>
<td>0.91</td>
<td>0.33</td>
<td>0.42</td>
<td>0.90</td>
<td>0.67</td>
</tr>
<tr>
<td>B</td>
<td>9</td>
<td>0.91</td>
<td>0.21</td>
<td>0.37</td>
<td>0.91</td>
<td>0.88</td>
</tr>
<tr>
<td>C</td>
<td>8</td>
<td>0.95</td>
<td>0.22</td>
<td>0.28</td>
<td>0.98</td>
<td>0.94</td>
</tr>
<tr>
<td>D</td>
<td>53</td>
<td>0.93</td>
<td>0.19</td>
<td>0.67</td>
<td>0.94</td>
<td>0.90</td>
</tr>
<tr>
<td>E</td>
<td>4</td>
<td>0.67</td>
<td>0.13</td>
<td>0.45</td>
<td>0.71</td>
<td>0.66</td>
</tr>
<tr>
<td>Overall</td>
<td>81</td>
<td>0.92</td>
<td>0.20</td>
<td>0.57</td>
<td>0.93</td>
<td>0.87</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>KPI stream ID</th>
<th>ADS</th>
<th>ROCKA + Opprentice</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\alpha)</td>
<td>0.86</td>
<td>0.62</td>
</tr>
<tr>
<td>(\beta)</td>
<td>0.91</td>
<td>0.20</td>
</tr>
<tr>
<td>(\gamma)</td>
<td>0.72</td>
<td>0.46</td>
</tr>
<tr>
<td>(\delta)</td>
<td>0.80</td>
<td>0.55</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
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.

• The anomaly detection results of ROCKA + Opprentice on KPI stream \( \alpha \), and \( \alpha \)'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 !