Project: Explaining anomalies in event stream monitoring

1. Project Description
This paper provides high-quality explanations for anomalous behaviours that users annotate on event monitoring results. Given the requirements for explanations, namely, conciseness, consistency with human interpretation, and prediction power, most existing techniques cannot produce explanations that satisfy all three of them. The key technical contributions of this work include a formal definition of optimally explaining anomalies in event monitoring, and three key techniques for generating sufficient feature space, characterizing the contribution of each feature to the explanation, and selecting a small subset of features as the optimal explanation, respectively.

2. Reading Material
Please read the paper from the beginning up to Section 5.4 to understand the relevant algorithms. The relevant concepts and techniques include the requirements on explanations (Section 2), feature engineering (Section 3), single-feature reward (Section 4), constructing explanations using multiple features (Section 5). Please also read Section 6 for the evaluation methodology.

3. Algorithms to Implement
Please implement the following algorithms:
Section 4: Single feature reward (Section 4)
Section 5: Constructing explanations (Section 5.1 and 5.3)

In particular, Section 5.2 includes techniques (“Identifying related partitions”, “Partition alignment” and “Interval labelling”) that need to be customized for our given dataset. They are not required for the project but do affect the quality of explanations returned. More details will be given when we discuss ideas for extension.

4. Results to Produce
Dataset: event streams will be provided by the course staff.

Results to produce:

- Fig 15 (Conciseness): Given some anomalies, please reproduce the plot for “XStream” and “XStream-cluster”. There is no need to produce other techniques for comparison.
Formally, we model each trace in the test dataset as a multi-dimensional time series, \([x_1 \ldots x_T \ldots x_m]^T\), where each data item includes \(m\) features, \(x_t = (x_{t1}, \ldots, x_{tm})\). A detected anomaly is a subsequence of the time series that starts at timestamp \(t\) and has duration \(w, X_{t,w} = [x_t \ldots x_{t+w}]^T\). We denote the explanation generated for the anomaly \(X_{t,w}\) as \(F_{t,w}\) and treat it as a function of the features, \(A = (a_1, \ldots, a_m)\), from the data:

\[
F_{t,w}(a_1, \ldots, a_m) = X_{t,w}
\]

where \(\Rightarrow\) means that \(F_{t,w}\) "explains" the anomaly \(X_{t,w}\). In addition, we define an extraction function, \(G_A\) over \(F_{t,w}\), that returns the set of features used in the explanation (e.g., appearing in a logical formula or having non-zero coefficients in a regression model).

\[
G_A(F_{t,w}(a_1, \ldots, a_m)) = \{a_i | a_i \not\in A\}
\]

Finally, we define the size of \(F_{t,w}\) as the size of its feature set \(A_{t,w}\):

\[
|F_{t,w}(a_1, \ldots, a_m)| = |A_{t,w}|
\]

**Stability** means that the anomalies occurring in a similar context (e.g., for the same application, same run, and same time period) should have similar explanations, subject to small perturbation of the data.

Formally, we introduce a subsampling procedure over an anomaly \(X_{t,w}\), which generates a set of samples, \(\{X_{t,w}^{(i)}\}\). We denote the corresponding explanations generated for them as \(\{F_{t,w}^{(i)}\}\). The extraction function on a set of explanations is defined to be the duplicate-preserving union (like `Union All` in SQL) of the extraction function of each respective explanation:

\[
G_A(\{F_{t,w}^{(i)}(a_1, \ldots, a_m)\}) = \bigcup_i G_A(F_{t,w}^{(i)}) = \bigcup_i A_{t,w}^{(i)} = A_{t,w}
\]

Finally, for each feature \(a_j \in A_{t,w}\), we count its frequency in this feature set and normalize it by the total size of the feature set.

\[
H(A_{t,w}^{(j)}) = -\sum_i p(a_j) \log_2 p(a_j), \quad a_j \in A_{t,w}^{(j)}
\]

where \(p(a_j) = \frac{1_{A_{t,w}^{(j)}}(a_j)}{|A_{t,w}^{(j)}|}\) is an indicator function that counts the occurrences of a feature in a multi-set. For capturing consistency, our choice of entropy is motivated by information theory that a set of explanations that lack consistency will require using more bits to encode, hence a larger entropy value. In the ideal case, all explanations, \(\{F_{t,w}^{(i)}\}\), are identical, and its entropy takes the minimum value 0 if the size of the explanation is 1 (denoted as \(H_1\)), the value 1 if the size is 2 (\(H_2\)), or the value 1.58 if the size is 3 (\(H_3\)). We consider an entropy value within \(H_3 = 1.58\) a good score for consistency.

5. **Ideas for Extension (20%)**

This project also requires an extension. We offer a few ideas below. Please feel free to **choose one idea to explore** in your project. Students are also welcome to propose their own idea.

1) **Write an anomaly detection algorithm and detect other anomalies in the dataset.**

2) **Filter incidentally correlated features.** It is known that the explanation construction algorithms may choose features that are incidentally correlated to the small set of labelled anomalous instances. For example, `timestamp` may be chosen to construct an explanation, but it is actually a bad feature to use because the explanation constructed using `timestamp` does not generalize to other anomalous instances. In particular, Section 5.2 includes techniques ("Identifying related partitions", “Partition alignment” and “Interval labelling”) that need to be customized for our given dataset. Please propose methods to adapt these techniques to our dataset in order to filter incidentally correlated features from the constructed explanations.