Semi-supervised anomaly detection with an application to water analytics

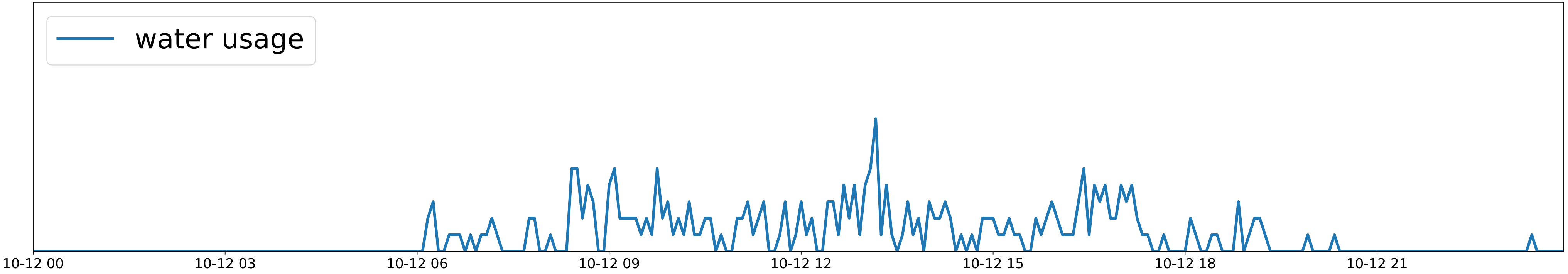
Vincent Vercruyssen[1]
Wannes Meert[1]
Gust Verbruggen[1]
Koen Maes[2]
Ruben Bäumer[2]
Jesse Davis[1]

1 KU LEUVEN
2 colruyt

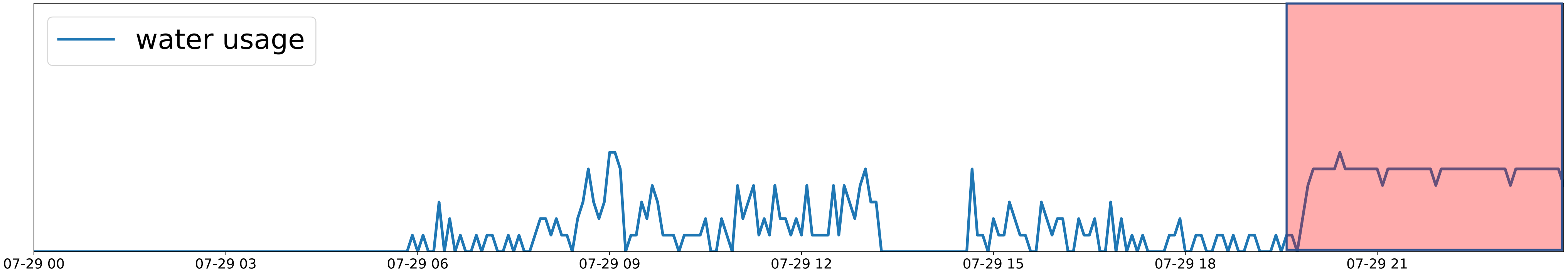
DTAI
Declaratieve Talen en Artificiële Intelligentie

Detecting abnormal water usage

- **GIVEN:** Colruyt Group records water consumption in retail stores

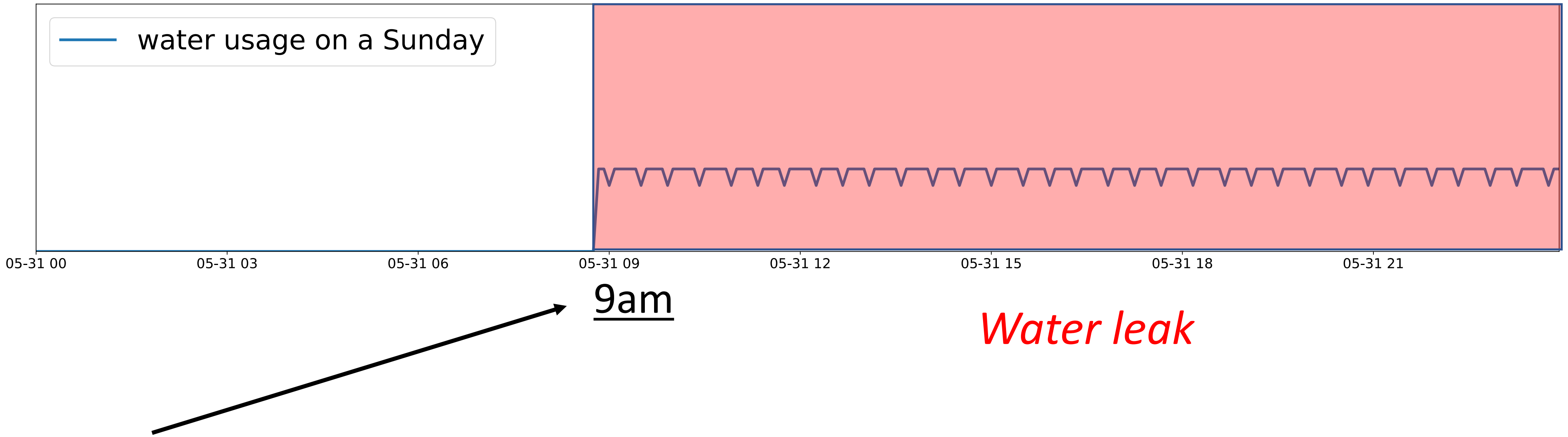


- **DO:** detect periods of anomalous water consumption in time series



Why automatic anomaly detection?

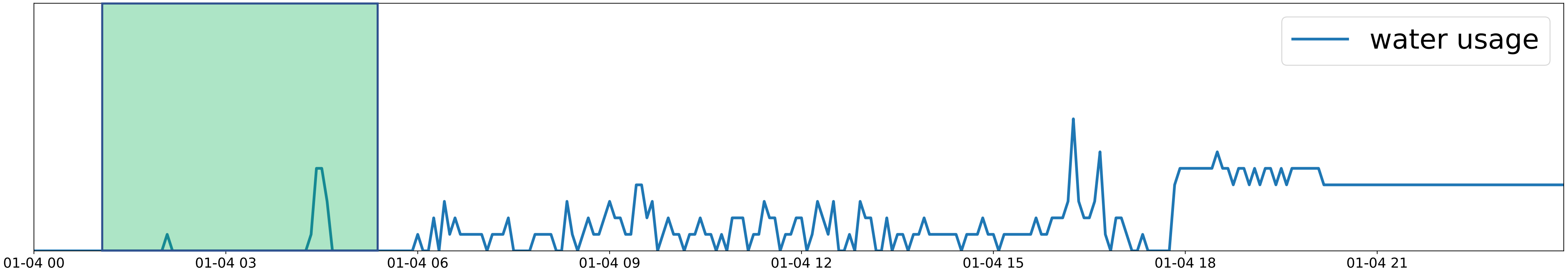
- Unusual water consumption in a store carries costs



- Early detection of anomalies helps reducing said costs

Challenge: infrequent normal behavior

- Colruyt currently employs a rule-based detection system
- BUT what about infrequent normal behaviour?



Normal behavior linked to maintenance (water softener)

Flagged as anomaly by rule-based system

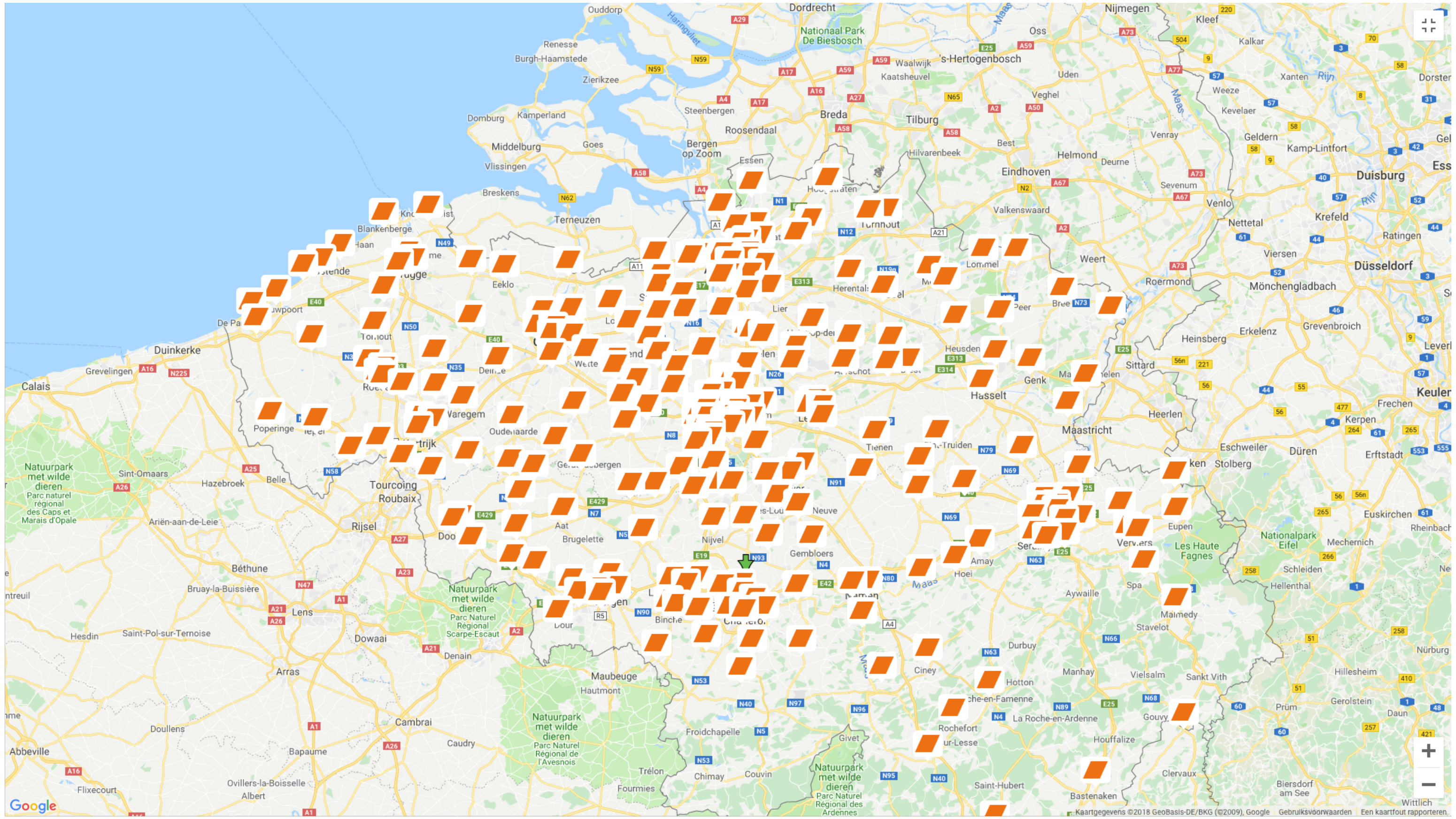
Challenge: heterogeneous signal

- BUT what about heterogeneous quality of the signal?

Both patterns are normal behavior! *Top normal, bottom anomaly!*

Challenge: large number of stores

- Colruyt operates hundreds of stores in Belgium



Typical approach: unsupervised learning

- **GIVEN**: a dataset without label information
- **DO**: construct a model to detect anomalies in the dataset

- Requires an assumption about anomalies to detect them
  - Geometric: anomalies are far away from normals
  - Statistical: anomalies lie in low-density region

✓ Requires no labels
✗ Strong assumptions about the nature of anomalies
✗ Difficult to evaluate

Typical approach: supervised learning

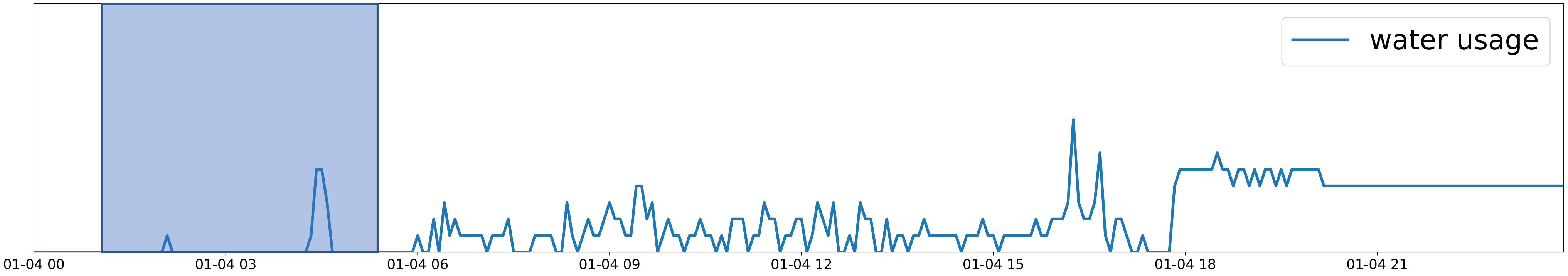
- **GIVEN**: a dataset with label information
- **DO**: construct a model to detect anomalies in the dataset

- Requires all data points to be labeled

✓ Generally good performance
✓ No assumptions about the nature of anomalies
✓ Easy to evaluate
✗ Requires a prohibitive number of labeled examples

Our approach:
semi-supervised learning & active learning

- Ground detection in the unsupervised learning paradigm
- Ask the expert for simple feedback to improve the model



Is this normal or anomalous behaviour?

Semi-supervised detection of outliers (SSDO)

1. Compute an initial unsupervised anomaly score for each data point

   → low-density regions of the instance space are anomalous

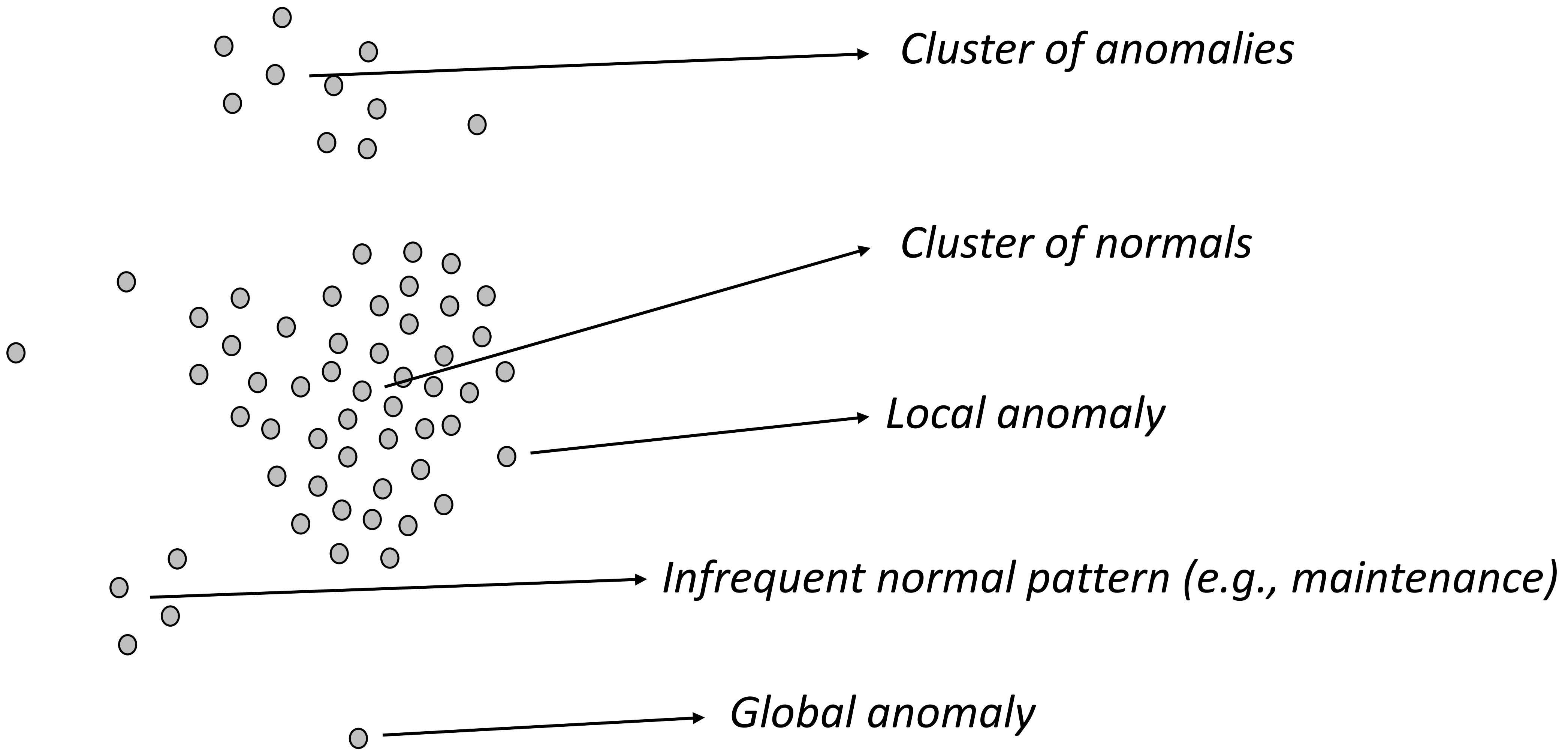
2. Propagate the known labels through the data

   → points that are close to each other likely have the same label

3. Active learning to query the user for additional label information

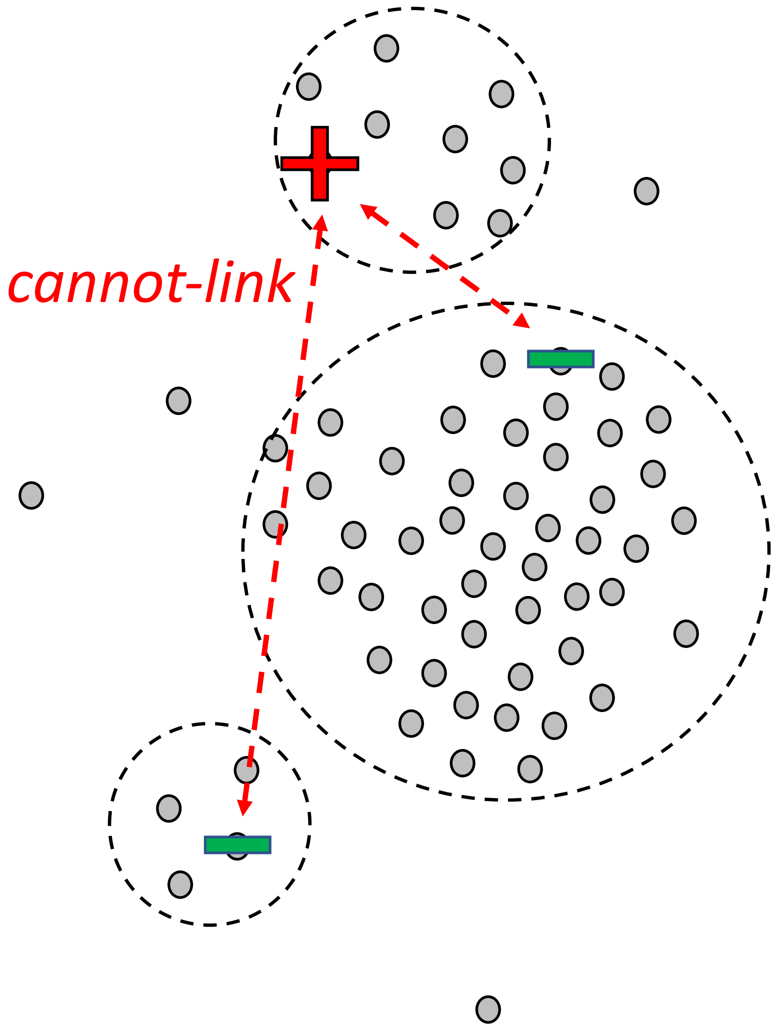
   → improve the model where it is uncertain

Toy example



Cluster of anomalies
Cluster of normals
Local anomaly
Infrequent normal pattern (e.g., maintenance)
Global anomaly

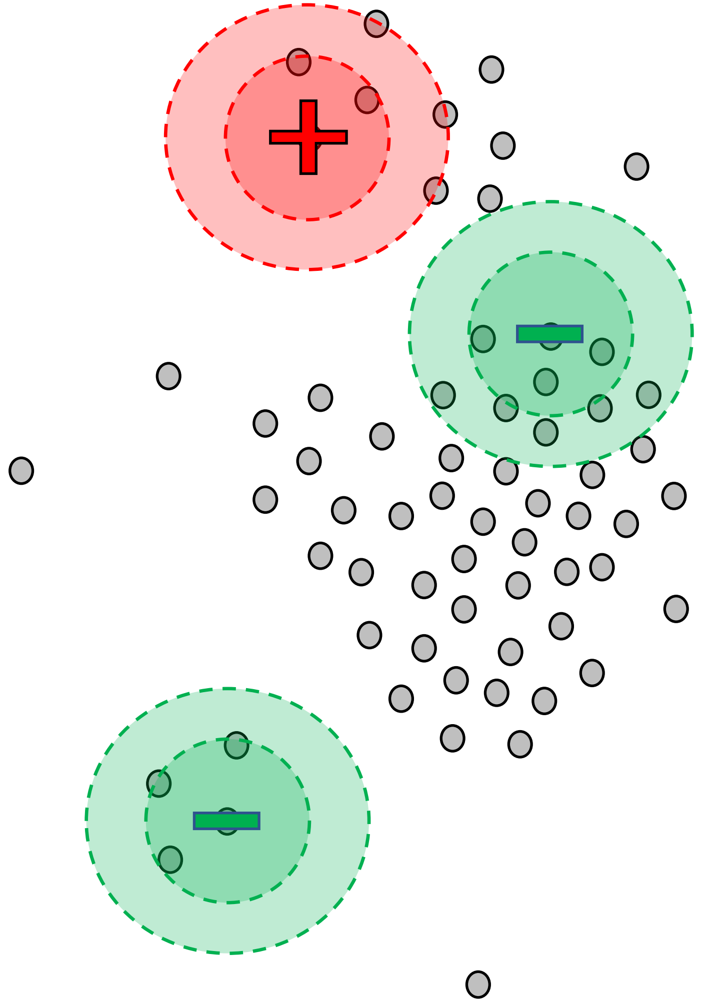
1. (Constrained) clustering score



- Cluster the data
- Use constraints to guide the clustering
- Derive the initial anomaly score for each point f

$$score(x) \sim \frac{point\_deviation(x) \times cluster\_deviation(x)}{cluster\_size(x)}$$

2. Label propagation



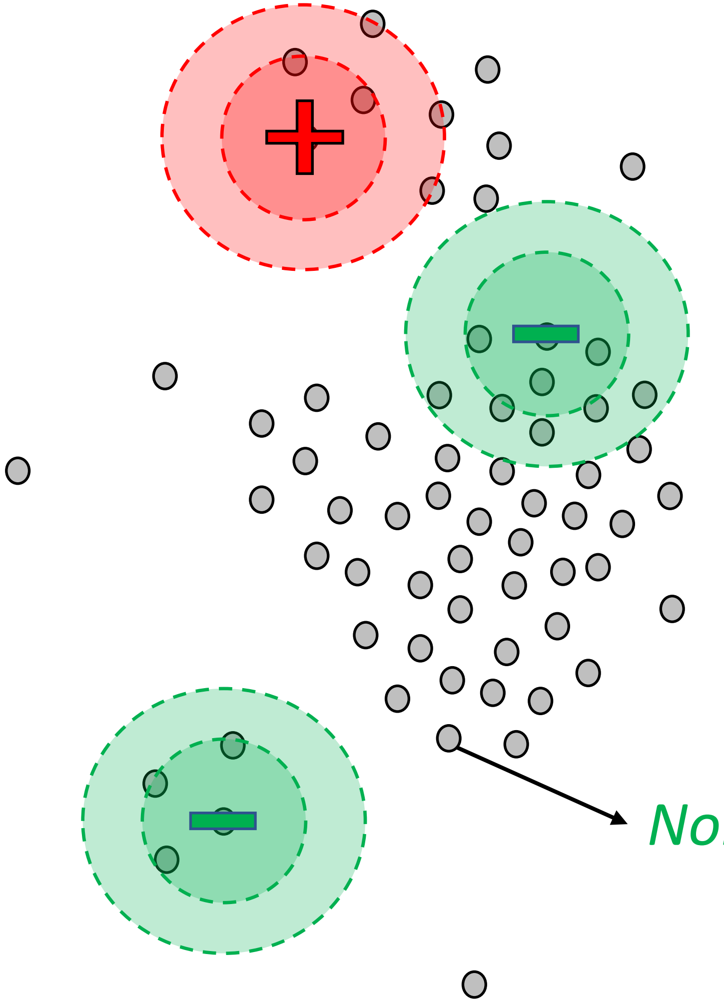
- Known labels are propagated through the data
  - 2 points close to each other likely share the same label
  - Labels influence near points more than far off points

- Derive the final anomaly score for each point x

$$S(x) = \frac{1}{Z}\left[\underbrace{score(x)}_{\textit{initial score}} + \alpha \underbrace{\sum_{x_i \in L_a} g(d(x, x_i))}_{\textit{label propagation score}}\right]$$

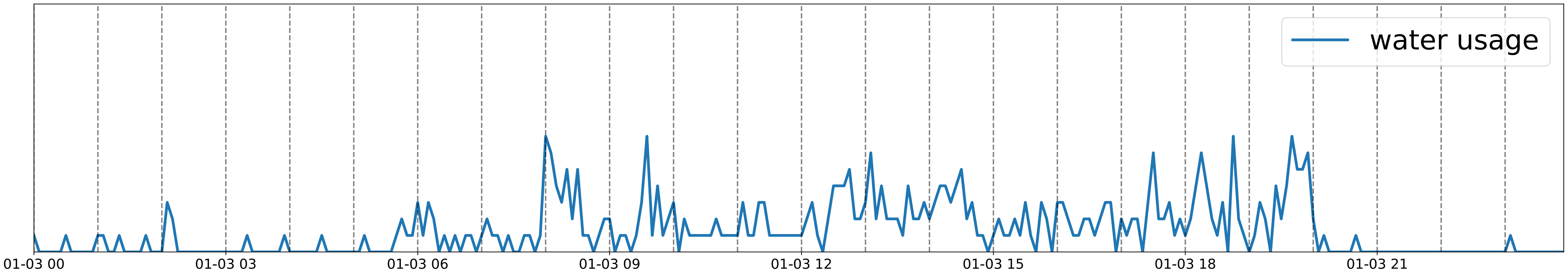
normalization

3. Active learning



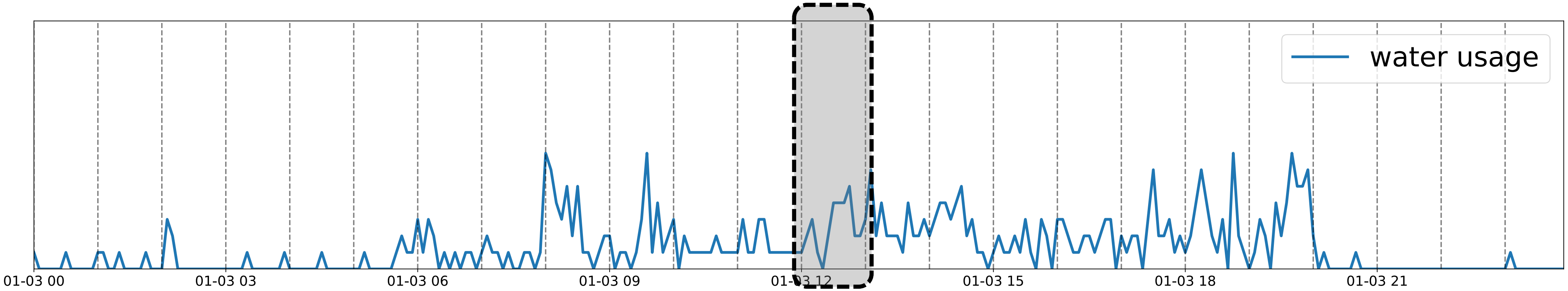
- Query the user for additional label information
- Uncertainty sampling queries the example with the least certain predicted label
- Diverse queries

Use SSDO for water usage data



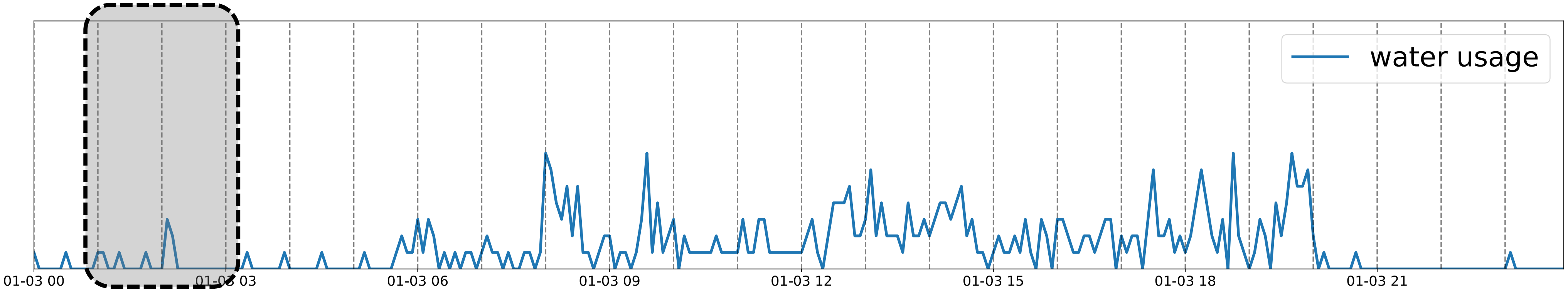
1. Divide time series into one-hour non-overlapping windows
   - Colruyt expert can only label one hour
   - Ease of communicating and interpreting anomalies
2. Transform each window to feature vector
3. Run SSDO on the feature vector representation
4. Go back to original time series to indicate anomalous periods

2. Transform window to feature vector

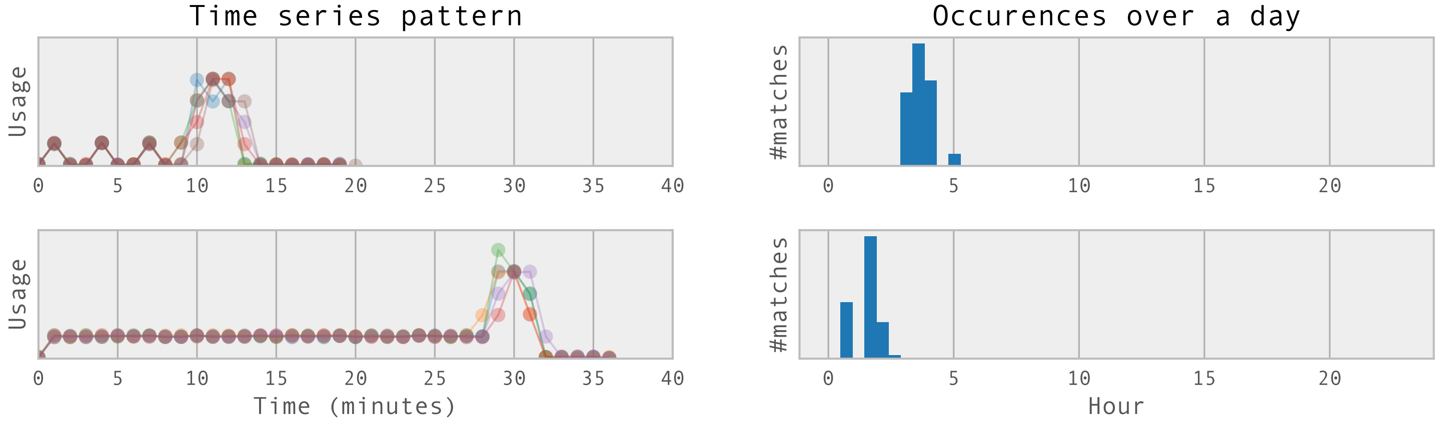


1. Time is an essential feature → *windows of one hour*
2. General behavior → *summary statistics*
3. Normal during week, not weekend → *time-of-day, day-of week…*

2. Transform window to feature vector



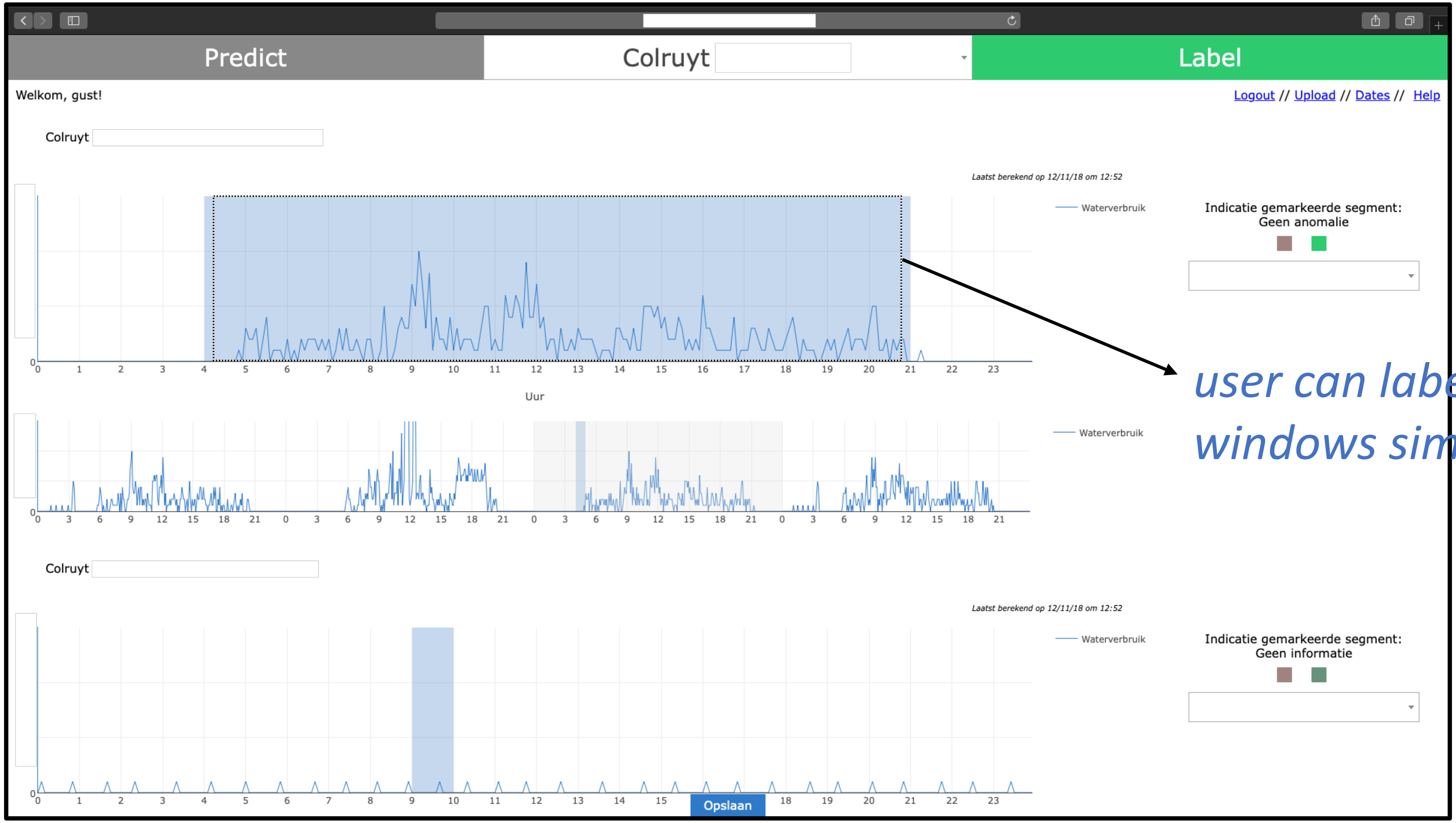
4. Fixed patterns → *shape features capturing the pattern*



3. SSDO predicts anomalies

web interface

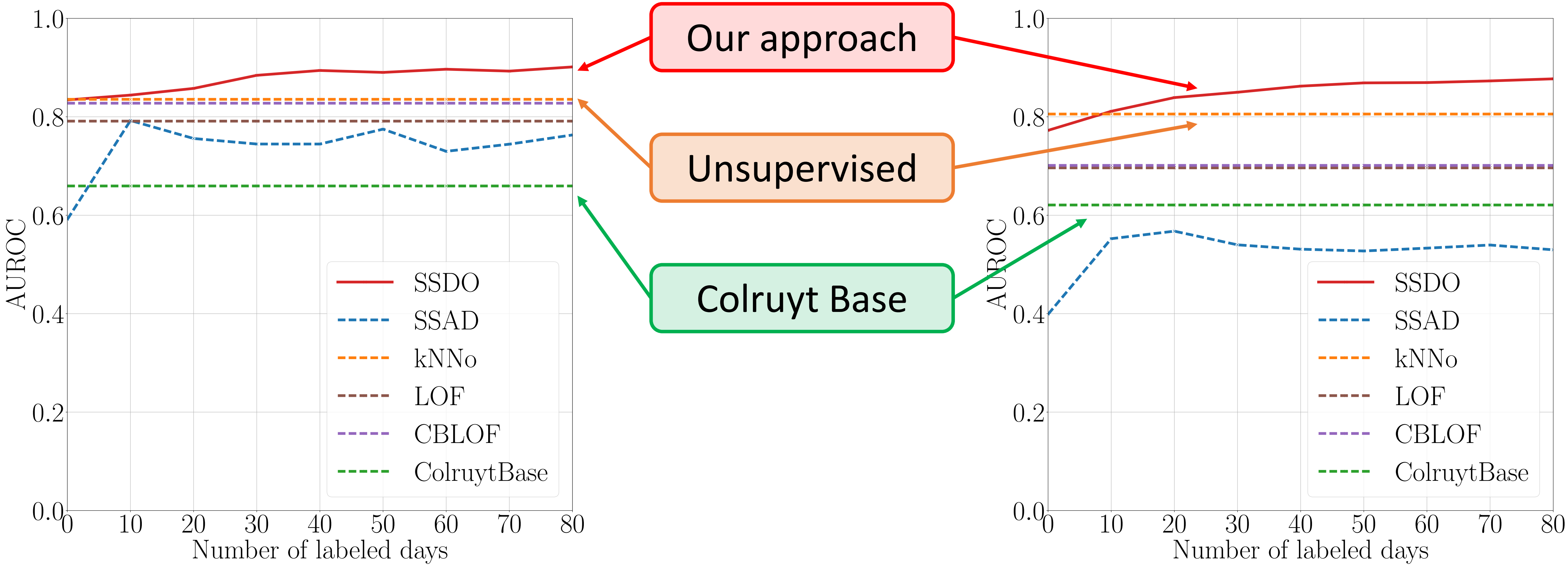
4. SSDO queries the user



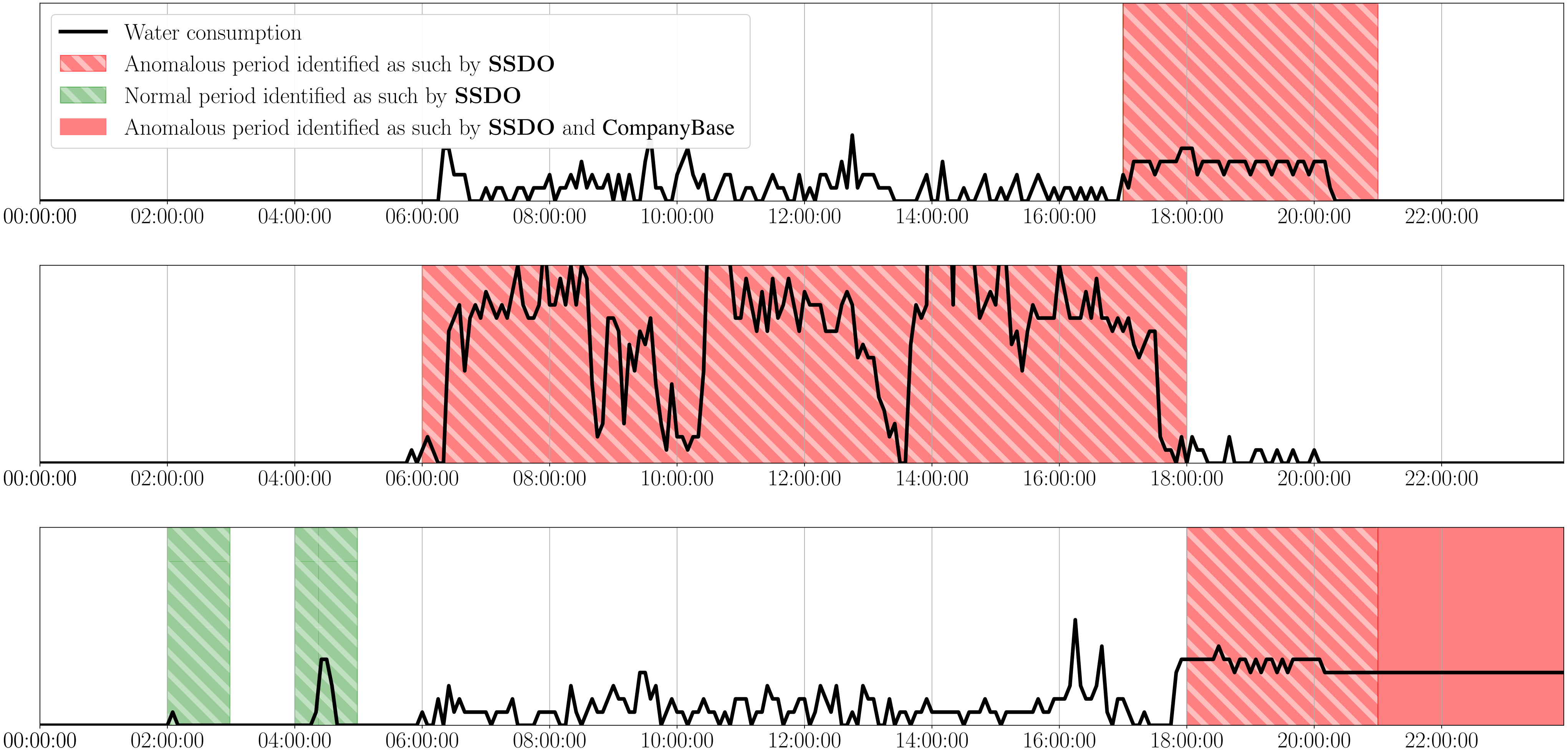
user can label multiple windows simultaneously

web interface

Results of the deployed system in 2 stores



Did we solve the challenges?



Take-away

1. Semi-supervised anomaly detection with an active learning loop
2. Accurately describing time series data requires a heterogeneous feature representation
3. Deployed system that outperforms existing approaches

Questions?