Semi-supervised Anomaly Detection with an Application to Water Analytics

Vincent Vercruyssen
Dept. of computer science
KU Leuven
Leuven, Belgium
vincent.vercruyssen@cs.kuleuven.be

Wannes Meert
Dept. of computer science
KU Leuven
Leuven, Belgium
wannes.meert@cs.kuleuven.be

Gust Verbruggen
Dept. of computer science
KU Leuven
Leuven, Belgium
gust.verbruggen@cs.kuleuven.be

Koen Maes
Colruyt Group
Halle, Belgium
koen.maes@colruytgroup.com

Ruben Bäumer
Colruyt Group
Halle, Belgium
ruben.baumer@colruytgroup.com

Jesse Davis
Dept. of computer science
KU Leuven
Leuven, Belgium
jesse.davis@cs.kuleuven.be

Abstract*—Nowadays, all aspects of a production process are continuously monitored and visualized in a dashboard. Equipment is monitored using a variety of sensors, natural resource usage is tracked, and interventions are recorded. In this context, a common task is to identify anomalous behavior from the time series data generated by sensors. As manually analyzing such data is laborious and expensive, automated approaches have the potential to be much more efficient as well as cost effective. While anomaly detection could be posed as a supervised learning problem, typically this is not possible as few or no labeled examples of anomalous behavior are available and it is oftentimes infeasible or undesirable to collect them. Therefore, unsupervised approaches are commonly employed which typically identify anomalies as deviations from normal (i.e., common or frequent) behavior. However, in many real-world settings several types of normal behavior exist that occur less frequently than some anomalous behaviors. In this paper, we propose a novel constrained-clustering-based approach for anomaly detection that works in both an unsupervised and semi-supervised setting. Starting from an unlabeled data set, the approach is able to gradually incorporate expert-provided feedback to improve its performance. We evaluated our approach on real-world water monitoring time series data from supermarkets in collaboration with *Colruyt Group*, one of Belgiums largest retail companies. Empirically, we found that our approach outperforms the current detection system as well as several other baselines. Our system is currently deployed and used by the company to analyze water usage for 20 stores on a daily basis.

***Keywords*-anomaly detection; time series; water analytics; semi-supervised machine learning**

I. Introduction

Modern organizations use a variety of sensors to continuously monitor equipment and natural resources. Often, the obtained measurements are stored centrally in order to enable detailed monitoring and swift interventions whenever a fault or suboptimal behavior is detected. Typically, automated strategies are needed to monitor the collected data because it is not feasible for a person to monitor hundreds of machines, plants or buildings simultaneously. Such strategies have been successfully applied in many organizations, for example by using condition monitoring methods [1]. Originally, anomalous behavior in time series was detected by statistical methods like exponentially weighted moving average, or cumulative sum [2]. Often, this approach is too simplistic for complex signals and as a result more elaborate formulas were handcrafted to differentiate between normal and abnormal behavior [3]. However, maintaining a formula-based system is nearly impossible as it needs to be modified whenever the monitored system changes. Paradoxically, even when confronted with a complex signal, many operators are able to discern patterns which indicate when a system is behaving abnormally. This insight has motivated applying machine learning to learn models of typical behavior by observing the system [4].

From a machine learning perspective, three challenges frequently arise in real-world anomaly detection tasks. First, it is difficult to treat anomaly detection as a supervised learning problem because few or no ground-truth examples of anomalous behavior are available. Furthermore, collecting all possible types of anomalies upfront is infeasible in practice. Nor is it desirable to permit anomalous behavior for the sake of generating data as such behavior may have an expensive and negative impact (e.g., result in breaking a machine). Second, while unsupervised methods do not require labels, they make strong assumptions that are violated in practice, which makes for a less robust anomaly detector. The most widely used assumption is that infrequent behavior is anomalous, which is not always true. For example, maintenance operations on a system can occur less frequently than the system wasting a resource. Consequently, some form of labeling and tuning is required in practice. Third, when analyzing the time series data, the signal requires a combination of heterogeneous descriptors. Three types of anomalies are typically considered: point-wise, contextual, and collective [5]. Point-wise anomalies are detected by stochastic descriptors and used when the evolution of the series is not predictable; collective anomalies can be detected by typical time series patterns such as shapes, motifs or residuals; and contextual anomalies by looking at the previous types in context of other information such as day of week. Ideally, a detection system should find all types of anomalies.

In this paper, we tackle the problem of automatically detecting periods of abnormal water consumption in a real-world setting at *Colruyt Group*, a large Belgian retail company ($>$400 stores). This use case exhibits each of the aforementioned machine learning challenges. Water consumption has a complex pattern as it is a combination of fixed behavior (e.g., maintenance, opening hours that differ daily), stochastic behavior due to external influences (e.g., restroom visits, number of customers in the store) and combinations of both (e.g., meat preparations, cleaning). Furthermore, several normal behaviors (e.g., maintenance) occur less frequently than some abnormalities, violating the assumption underlying unsupervised anomaly detection. As part of *Colruyt*'s policy for sustainable entrepreneurship, they employ energy monitoring systems that track the consumption of electricity, water, gas, etc. in their stores, office buildings, and distribution sites. Reducing water usage has clear environmental benefits. For instance, 11% of Europe's population live in regions that are water stressed. Because of losses in the water supply network, as much as 30% of water is lost before it reaches the consumer in countries such as France and Spain [6]. Droughts have significantly impacted Europe in the past decades and this situation is expected to deteriorate [7]. Early detection of leakages, and malfunctions could help prevent these losses.

This paper proposes a novel semi-supervised approach to anomaly detection and discusses how to apply it to water consumption monitoring. To address the first two challenges, the method starts out as an unsupervised model that uses a clustering-based approach to define typical normal behavior and identify anomalies. Our system employs active learning so that a domain expert can provide feedback to the model in the form of ground-truth labels. Whenever feedback is provided, the approach operates in a semi-supervised manner and exploits the labels in two ways. One, the labels are used to guide the clustering in the form of constraints. Two, the initial cluster-based anomaly score is updated via a label propagation step to maximize the usefulness of the acquired labeled data. Thus, initially no labels were available but the method gradually improves with feedback and is increasingly able to distinguish between more subtle differences in behavior. To apply this to water monitoring, we generate a set of features that describe different characteristics of the signal (e.g., statistical properties, motifs) as well as contextual variables (e.g., day of week) in order to capture the heterogeneity of the time series signal. Empirically, our approach outperforms several well-known competing anomaly detection methods as well as the currently used baseline at *Colruyt* on real-world water consumption data provided by *Colruyt Group*. Currently, our system is deployed and used to monitor water usage in 20 *Colruyt* stores and we report on insights gleaned from its daily operation. To summarize, this paper makes the following contributions:

- A novel combination of machine learning methods to design a semi-supervised approach to anomaly detection;
- A description of how to apply our approach to the water consumption use case;
- An empirical study that shows the benefits of our proposed approach compared to existing anomaly detection approaches as well as the company's baseline approach; and
- A discussion of the deployed system architecture, which is used on daily basis, and illustrative examples of its real-world performance.

II. Related Work

Interest in developing automated anomaly detection methods has rapidly increased in recent years, with surveys appearing covering: anomaly detection [5], novelty detection [8], and outlier detection for temporal data [9]. We now highlight the most closely related use-case specific and algorithmic work.

A. Natural Resource Analytics

For water analytics, research has been directed mainly towards detecting anomalies in water distribution, or water quality over time [10]. Fagiani et al. [11] compare several unsupervised methods (GMM, HMM, and one-class SVM) for leakage detection in water and natural gas grids. Patabendidge et al. [12] applied the k-nearest neighbor outlier detection method (kNNo) to find outliers in water usage data from aquatic leisure centers [13]. That work uses a much more simplistic set of statistical features to characterize the time-series signal and can therefore not be applied directly on the case presented in this work. The present paper contains an extensive empirical evaluation comparing to multiple baselines, including kNNo, but with more complex features. Furthermore, the models mentioned in the above related work do not have the ability to take expert feedback into account.

More research has studied anomaly detection for the consumption of other natural resources, specifically (household) electricity consumption. Jakkula and Cook [14] compare statistical with unsupervised clustering-based techniques for detecting periods of unexpected consumption. Janetzko et al. [15] employ both a straightforward load-forecasting technique and an unsupervised clustering-based technique for outlier detection in electricity consumption data. Chou and Telaga [16] construct a hybrid neural net ARIMA model to predict electricity consumption, and flag anomalies that deviate from the prediction using the two-sigma rule. Contrary to all of the above, our approach works in both the unsupervised and semi-supervised setting.

B. Unsupervised Anomaly Detection

The typical assumption in the unsupervised setting is that outliers represent potential anomalies. Density-based outlier detection techniques compute the density of the data distribution around data points to find anomalies. Breunig et al. [17] proposed the *local outlier factor* (LOF), an absolute outlier score equaling the ratio of the average density of the k-nearest neighbors of a data point and the local density of the point itself. Numerous variations on this idea have been proposed [18]–[20]. In contrast, we calculate an anomaly score using a constrained-clustering-based approach that is able to incorporate label information.

He et al. [21] proposed the *cluster-based local outlier factor* (CBLOF). Like our approach, CBLOF is a clustering-based approach. A key difference between our approach and CBLOF is that the latter is purely unsupervised. In an unsupervised setting, our score differs from CBLOF in two distinct ways. First, CBLOF requires two additional parameters to partition the set of clusters into large and small clusters. Second, unlike CBLOF, our score also considers the distance between clusters as a factor of anomalousness.

Finally, Ramaswamy et al. [13], proposed the k-nearest neighbors outlier detection method (kNNo). Each point's anomaly score is the distance to its k^{th} nearest neighbor in the data set. Then, all points are ranked based on this distance. The higher an example's score is, the more anomalous it is. A recent extensive empirical evaluation of unsupervised outlier detection algorithms found that kNNo and LOF perform exceedingly well in practice [22].

C. *Semi-supervised Anomaly Detection*

Even though exploiting label information in the anomaly detection task has clear benefits, only a few semi-supervised anomaly detection algorithms exist. Görnitz et al. [23] frame anomaly detection as an optimization problem known as *support vector data descriptions* and propose a generalization that processes both labeled and unlabeled examples. Das et al. [24] propose a semi-supervised approach that employs active learning. It uses the unsupervised ensemble-based Loda anomaly detection algorithm [25] in combination with a semi-supervised extension of the accuracy-at-the-top loss function to update Loda's ensemble weights based on the user's feedback. Its active learning strategy is to select the most anomalous unlabeled example for labeling.

III. Our Proposed Approach

We now describe SSDO (Semi-Supervised Detection of Outliers), a novel, general, semi-supervised framework for anomaly detection. The next section will discuss the details of applying SSDO to the water consumption monitoring use case. SSDO works on data $F = \{f_0, \ldots, f_n\}$, with each example f_i represented in the standard feature-vector format. It assigns an anomaly score to each example in a two-step process. First, an initial score is assigned based on an example's position with respect to the data distribution found using constraint-based clustering. Second, if some examples have known labels, then there is a label propagation phase. Each example's anomaly score is updated based on nearby examples' known labels. The approach employs an active learning strategy to acquire more labels, and the process is repeated whenever additional data or labels are provided.

A. *Constrained Clustering*

The first step of SSDO clusters the data. We employ clustering because it allows deriving an anomaly score in a purely unsupervised manner if no labels are provided. However, if labels are available, we are able to incorporate them to guide the clustering in the form of constraints. We do so by including one cannot-link constraint between each anomalous and normal example. We do not consider must-link constraints, because there is no guarantee that either all normal behavior or all abnormal behavior is similar. For example, there can be different types of anomalies and different anomaly types should not be forced to appear in the same cluster. We employ the COP k-means constrained clustering algorithm [26], which reduces to standard k-means if no constraints are provided. Note that any constrained clustering algorithm that defines a centroid for each cluster could also be used.

A standard assumption in unsupervised anomaly detection is that anomalies look "different" from normal examples. From a geometric perspective, this means they will be far away from normal examples. From a statistical perspective, this means they will lie in a low-density region of the instance space. Following this assumption, the following three intuitions about a clustering could be helpful in identifying anomalies: (1) the more an example deviates (i.e., is far away) from its cluster centroid, the more likely it is to be anomalous, (2) the more a cluster centroid deviates from the other cluster centroids, the more likely it is to represent anomalies, and (3) the smaller a cluster is, the more likely it is to contain anomalies.

Based on these intuitions, the clustering is used to assign each example $f \in F$ the following anomaly score:

$$score(f) = 1 - g\left(\frac{point\_dev(f) \times cluster\_dev(f)}{cluster\_size(f)}; \gamma\right) \tag{1}$$

where each of its three components are defined next.

The **point deviation** captures the deviation of f from its cluster center, normalized by the maximum deviation from the center over all data points within its cluster:

$$point\_dev(f) = \begin{cases} \frac{d(f,\ c(f))}{\max_{f_i \in C(f)} d(f_i,\ c(f))} & \text{if } |C(f)| > 1 \\ 1 & \text{otherwise} \end{cases} \tag{2}$$

where $C(f)$ represents all the examples in the same cluster as f, $d(.)$ is the (Euclidean) distance function, and $c(f)$ is the centroid of the cluster f belongs to. If $C(f)$ contains a single example, the point deviation is 1.

The **cluster deviation** measures how much cluster centroid $c(f)$ differs from the other cluster centroids weighted by the maximum inter-cluster distance:

$$cluster\_dev(f) = \begin{cases} \frac{\min_{1 \leq i \leq n_c \wedge c(f) \neq c_i} d(c(f),\ c_i)}{\max_{1 \leq i,j \leq n_c} d(c_i,\ c_j)} & \text{if } n_c > 1 \\ 1 & \text{otherwise} \end{cases} \tag{3}$$

where c_i represents the centroid of cluster i, and n_c is the number of clusters. If there is only one cluster, the cluster deviation is 1.

The **cluster size** expresses the size of the cluster f belongs to relative to the largest found cluster:

$$cluster\_size(f) = \frac{|C(f)|}{\max_{1 \leq i \leq n_c} |C_i|} \tag{4}$$

where C_i is the set of examples assigned to cluster i.

The **squashing function** g is used to map the anomaly score into the range between 0 and 1, with a higher value representing a greater degree of anomalousness. We use a squashing function from the exponential family:

$$g(x; \gamma) = 2^{-\frac{x^2}{\gamma^2}}. \tag{5}$$

The parameter γ is set such that the percentage of examples that have a score greater than 0.5 after mapping, is equal to the user's requested percentage of examples that the system should flag as anomalous if no labels are available.

Compared to CBLOF this approach requires less parameters and also considers the distance between clusters (see also Section II).

B. Updating the Scores via Label Propagation

In order to maximize the value of having labels, we update the initial anomaly score assigned by the clustering by combining it with a score obtained by performing label propagation. The updated score obtained for each example f is:

$$S(f) = \frac{1}{Z(f)} \left[score(f) + \alpha \sum_{f_j \in L_a} g(d(f, f_j); \eta) \right] \tag{6}$$

$$Z(f) = 1 + \alpha \sum_{f_j \in L} g(d(f, f_j); \eta) \tag{7}$$

where L_a is the set of labeled anomalous examples and L is the set of all labeled examples. For the squashing function $g(\cdot; \eta)$, η is the harmonic mean of the distance of each $f' \in F$ to its k^{th} nearest neighbor (i.e., k-distance) [17]. We use the harmonic mean because it is less sensitive to noise in the data. The goal of g is to make the influence of the label propagation depend on the distance between a labeled and unlabeled example. That is, a labeled example will have more effect on the score of closeby examples than on the score of far away examples. $Z(f)$ is a normalization factor such that the final score is between 0 and 1. Finally, α controls the impact of the user labels versus that of the clustering step. If α is larger than 1, the clustering is overruled immediately whereas if it is smaller, multiple labels are needed to overrule the clustering. Controling the labels' influence is desirable in practice because the user may incorrectly label examples, resulting in noisy labels [27]. Together, α and the chosen k-distance determine the speed with which the constraint-clustering-based model can change its mind (see Section V).

C. Querying for Labels

As previously mentioned, in the water consumption use case there are several types of normal consumption patterns that occur less often than anomalous patterns. It is difficult to distinguish between infrequent normal behavior and anomalies in a purely unsupervised manner. Therefore, acquiring labels offers the potential to improve performance. To do so, we employ the commonly used *uncertainty sampling* active learning framework [28]. In uncertainty sampling, the algorithm selects the example with the least certain predicted label and asks the user to label that example. In our setting, this entails selecting the example whose anomaly score is closest to 0.5.

IV. Anomaly Detection for Water Consumption

Colruyt is a retail group operating a large number of retail stores. We can formally define the task addressed in this paper as follows:

Given: A time series $T = \{(t_1, v_1), \ldots, (t_n, v_n)\}$, where t_i is a timestamp and v_i is the water consumption at time t_i, for a single retail store.
Do: Identify periods of anomalous consumption in T.

A. Data

The provided time series data report the water consumption over the last three years for 20 *Colruyt* stores. In each store, the consumption is measured every five minutes. There are no missing or incorrect measurements. Initially, none of the data was labeled.

B. Modeling Granularity

We perform anomaly detection on a store-by-store basis, and within each store we further subdivide the data T into non-overlapping windows of one hour (i.e., 00:00–01:00, 02:00–03:00, etc.). Working on the store level is important because each store has a different set of characteristics such as size, location, what services are offered, and opening hours, all of which affect water consumption. Larger stores have more employees and hence use more water. A butchery uses a large amount of water, however, not all stores offer this service. Furthermore, each store has a maintenance system, which is specifically configured for that store.

We divide the data within a store into the non-overlapping one hour windows for three reasons. First, and most importantly, this period was chosen because it is the smallest unit in which enough data is available for domain experts to detect typical patterns and label data. Second, time of day is an important feature that is implicitly captured by this partition. Finally, the anomalies have to be communicated to an expert, who may be monitoring multiple stores, and varying or "non standard" (e.g., 12:13–13:51) partitions reduce interpretability.

C. Feature Construction

Since describing the data requires heterogenous descriptors, each one hour window is transformed into a feature vector that describes the characteristics of the signal during that particular window. The following classes of features are constructed:

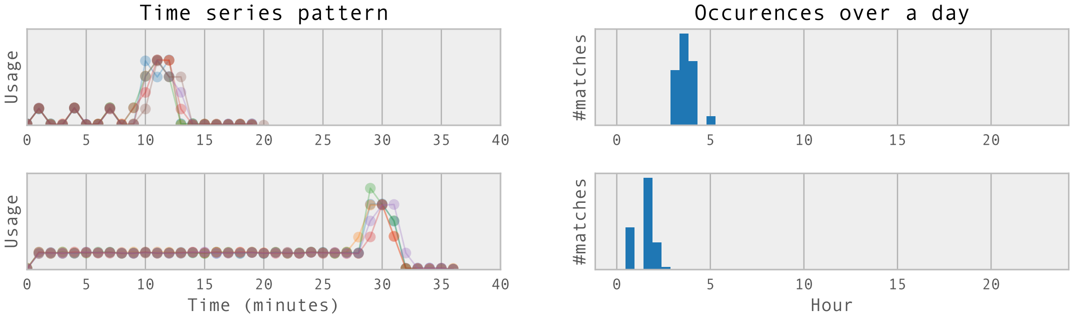


Fig. 1: Two patterns resulting from time series clustering. Subtly varying sequences are clustered together. A shape feature is the prototype of such a cluster. Both examples shown, match typical maintenance operations.

a) Summary statistics: We compute the following eight statistical features: max, min, mean, median value, standard deviation, skewness, kurtosis, and entropy.

b) Descriptive features: We consider three categorical features: the day of the week, the month of the year, and whether a day is a Belgian holiday or not. These categorical features are dealt with using one-hot encoding.

c) Shape features: The data contain certain operations, such as the maintenance, that have a typical usage pattern. While these operations typically begin around the same time, the number of days between successive operations is not fixed. Recognizing and reasoning about these shapes is important. To automatically identify these interesting shapes, a time series clustering approach is taken.[1] Because the duration of interesting usage patterns is unknown a priori, the data is coarsely discretized and given to the Prediction by Partial Match (PPM) algorithm [29] to efficiently detect coarse patterns that appear at least two times. This algorithm removes all the random sequences and leaves us with candidate patterns. Next, we apply time series clustering to all subsequences in the original data that correspond to a candidate coarse pattern [30]. The distance measure used for clustering, dynamic time warping, compensates for subtle variations introduced by the sensor's sampling strategy. Additionally, for each cluster we compute the histogram of occurrences throughout the day. We consider all the shapes that are a prototype of a cluster with at least five elements and where the entropy of the histogram is less than two to prefer patterns that are localized in time. For each of these shapes, there is one feature whose value is the dynamic time warping distance from the observed data to the shape. Fig. 1 shows an example of two patterns from two different stores that were found and typify a maintenance operation.

D. Applying SSDO to Water Consumption Data

The feature-vector representation of the water consumption data works as-is with the SSDO algorithm, given two important design choices. First, within each store, we construct 24 window groups (one for each one hour interval) and detect anomalies separately in each window group. The grouping is motivated by the importance of time of day as a factor governing the water consumption (e.g., the opening hours of a store affect consumption). Second, the active learning aspect functions slightly differently in practice due to the temporal nature of the data and its partition into hour long windows. When querying the user, we display a plot showing the water usage during the entire day (from 00:00 to 24:00) as well as the usage during the previous two days, because the context (e.g., time of day, day of week, etc.) is important to help the user determine the label. We highlight the hour of interest as selected by the active learning strategy. However, the user may select any contiguous block of time to label as either normal or anomalous. Hence the user can provide feedback on multiple windows simultaneously, which means the interface can naturally cope with anomalies spanning several windows. Labeling is always done on the level of a window, so by construction anomalies are always a multiplum of the window length. This is analogous to multi-instance learning: if any behavior in a window is anomalous, then the window is anomalous. Fig. 5 illustrates this process.

V. Experimental evaluation

We report the results on the evaluation of our algorithm as it is currently in use by *Colruyt*. We answer three questions:

Q1: How does SSDO perform compared to unsupervised and other semi-supervised anomaly detection algorithms?

Q2: How do the label propagation parameters α and k affect SSDO's performance with increasing user feedback?

Q3: How does each feature category contribute to the overall predictive performance?

A. Methods

Our empirical evaluation compares following approaches:

- **ColruytBase** is a set of hand-coded rules encoding expert knowledge *Colruyt* used to detect anomalous water consumption: if a rule is violated, the system flags an anomaly.[2]

[1] Implementation available: https://github.com/wannesm/dtaidistance

[2] While *Colruyt* employs advanced methods, namely fully supervised regression-based and load-forecasting algorithms, for monitoring other resources, applying these techniques to water consumption was challenging due its characteristics (e.g., infrequent non-anomalous patterns, stochastic behavior during the day).

- **kNNo** is a distance-based unsupervised outlier detection technique (in other work also referred to as kNN) [13].
- **LOF** is a density-based unsupervised outlier detection technique [17].
- **CBLOF** is a cluster-based unsupervised outlier detection technique [21].
- **SSAD** is a state-of-the-art semi-supervised anomaly detection approach based on one-class SVM [23].[3]
- **SSDO** is our method as outlined in Section III.[4]

These baselines were selected based on recent extensive empirical evaluations of outlier detection algorithms (see also Section II). Campos et al. show that, in practice, kNNo and LOF perform exceedingly well [22]. In particular, kNNo has achieved good results for water monitoring [12].

B. Data and Experimental Setup

a) Data: Colruyt experts have been labeling windows in 10 of the 20 stores over the past year. In each store we group the windows associated with the same hour. For the evaluation in this paper, we consider only the groups containing ≥ 50 labeled windows. This means there are labels for 50 unique days in each window group. However, which particular days are labeled varies by window group because the expert does not have to label all hours in a day. Four stores have window groups that fulfill this requirement with these stores having 24, 24, 23, and 10 groups, for 81 groups in total.

b) Experimental setup: The goal of the experiment is to evaluate how the anomaly detection method performs as more data is labeled. We perform the following experiment for each of the four stores separately. First, within each window group, we split the labeled windows into train set D_{train} and test set D_{test} using a stratified 50/50 split. Second, the labels in D_{train} are observed incrementally, which simulates beginning in an unsupervised setting and an expert gradually providing labels via the user interface (see Section 6.2). We initialize $D'_{train} = \varnothing$ and iterate over these three steps: (i) For each window group, run the anomaly detection algorithm over all examples in the group using the labels in D'_{train}; (ii) Evaluate the results on the test set D_{test}; (iii) Select 10 labeled days in accordance with the active learning strategy and add the labels to D'_{train}.[5] These steps are repeated until D'_{train} contains all the labels in D_{train}. For each method, we pick the parameter setting that maximizes training set area under the ROC curve (AUROC) in the final iteration. Note that this means that the unsupervised techniques use all the labeled examples in D_{train} to set their parameters. Finally, every experiment is repeated 20 times, with different, randomized train-test splits, and the results are averaged over the 20 runs.

c) Evaluation metrics: In step (ii), the anomaly detection algorithm outputs a ranking over the over the windows in D_{test} from most to least anomalous. We evaluate these rankings by computing the area under the ROC curve, as is standard in anomaly detection [22], [31]. Additionally, we define $\text{RE}_{x\%}$ as the effort required by the user to inspect x percent of the total number of anomalies in the test set, going through the ranking. For instance, if n equals x percent of the total number of anomalies in the test set, and the user needs to inspect only the first n consecutive instances in the ranking to find n anomalies, $\text{RE}_{x\%}$ is 1. If she needs to inspect $2n$ instances, $\text{RE}_{x\%}$ is 2, and so on. Lower values of $\text{RE}_{x\%}$ mean that the user needs to inspect fewer examples in the ranking to find $x\%$ of the anomalies.

d) Hyperparameters: For each method parameter, tuning is performed on the training data using grid search. ColruytBase has no parameters. LOF and kNNo each have one parameter k: the number of nearest neighbors, which is optimized over the interval $[1, 30]$. CBLOF has three parameters: the number of clusters n_c which is optimized over the interval $[1, 30]$, and the two parameters required to partition the set of clusters into large and small clusters, which are set to 90% and 5, as in the original paper [21]. SSAD has three parameters: the trade-off parameter κ is set to 1 as in the original paper, and the regularization parameters η_u and η_l are picked from $\{0.01, 0.1, 1, 10, 100\}$. We use SSAD with a Gaussian kernel. SSDO has four parameters: the number of clusters n_c is picked from the set $\{5, 10\}$, k is picked from the set $\{4, 16, 32, 64\}$, α is picked from the set $\{1, 1.5, 2.3, 4\}$, and the expected percentage of anomalies in the data is fixed to 5%.

C. Results

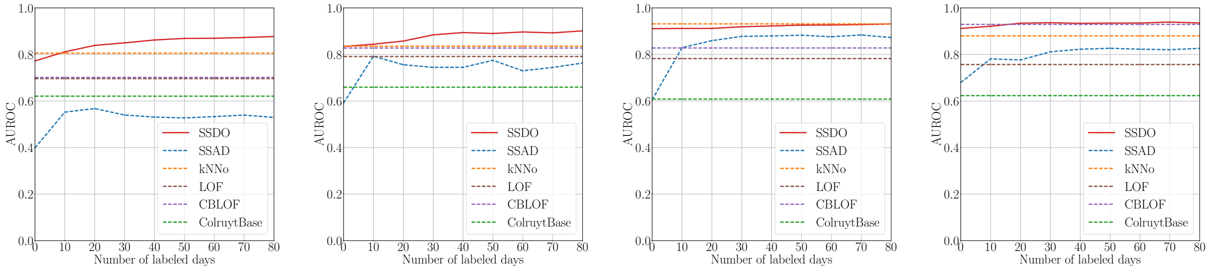
a) ***Q1:*** *Model performance:* Fig. 2 shows how the test set AUROC, averaged over all window groups per store, varies as a function of the number of labeled days for each method. Both SSDO and SSAD benefit from adding labels. However, SSDO achieves a much higher AUROC than SSAD in the unsupervised setting and SSAD is never able to approach SDDO's performance. On average, SSDO's AUROC improves by 1.8 percent after adding 10 labeled days, 5.6 percent after 40 labeled days, and 6.6 percent after 80 labeled days. In aggregate, SSDO outperforms SSAD, LOF, and ColruytBase, regardless of the number of labels provided. SDDO outperforms kNNo on three stores and achieves equivalent performance on the fourth. In fact, on three stores SSDO achieves equivalent or better performance than kNN with very few (i.e., 10 labeled days) or even no labels. SSDO also outperforms CBLOF by a wide margin on three stores and a small margin on the fourth.

Table I shows the average AUROC ranks over all 81 window groups for each method. We apply the hypothesis tests suggested by Demsar [32]. The Friedman test on the AUROC ranks rejects the null-hypothesis that all methods perform the same for both 0 and 80 labeled days (p-values < 0.001). With p = 0.05, the Nemenyi post-hoc test on the ranks finds that SDDO is significantly better than SSAD, LOF, and ColruytBase with no labels. After 80 labeled days, SSDO is also significantly better than CBLOF. Only kNNo remains competitive, albeit narrowly, as the difference in average ranks

[3] Implementation available: https://github.com/nicococo/tilitools

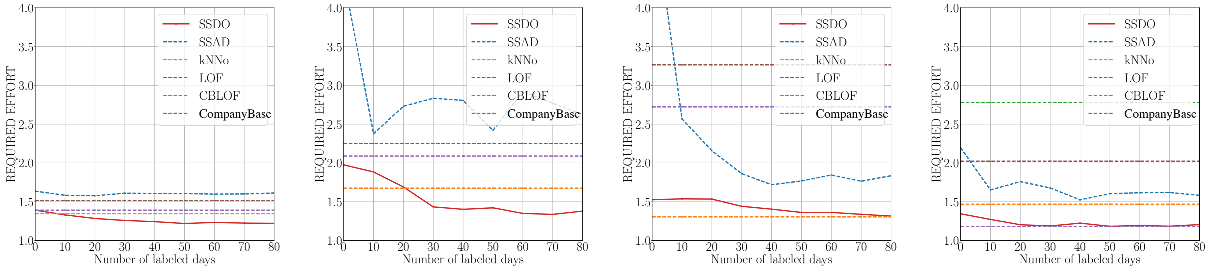
[4] Implementation: https://github.com/Vincent-Vercruyssen/anomatools

[5] Not all windows in the selected days have to be labeled, some days might only contain a few labeled windows.



(a) Average AUROC store 1. (b) Average AUROC store 2. (c) Average AUROC store 3. (d) Average AUROC store 4.

Fig. 2: Each store's average test set AUROC as a function of the number of days labeled by the user, shown for each method. SSDO consistently performs better than SSAD. SSDO also outperforms the unsupervised baselines in 3/4 stores, and ties with kNNo for first place in the last.



(a) Required effort store 1. (b) Required effort store 2. (c) Required effort store 3. (d) Required effort store 4.

Fig. 3: Each store's average test set $RE_{95\%}$ as a function of the number of days labeled by the user, shown for each method. In plots (b) and (c), $RE_{95\%}$ for ColruytBase is > 4. SSDO outperforms SSAD, LOF, and ColruytBase in each store. In each store, adding user feedback leads to a reduction in $RE_{95\%}$ for SSDO.

of kNNo and SSDO is slightly below the critical difference of the Nemenyi test.[6]

Fig. 3 shows how the test set $RE_{95\%}$, averaged over all window groups per store, varies as a function of the number of labeled days for each method. SSDO consistently outperforms SSAD, both with and without labels. Again, SSDO is better than kNNo in three stores and ties for first place in the last. SSDO's required effort is also substantially lower than CBLOF's in three stores and equal in one. On average, SSDO benefits from user feedback as labeling 10 days reduces the required effort by 3.4 percent while labeling an additional 30 days results in a 14 percent reduction.

Table II shows the average $RE_{95\%}$ ranks over all 81 window groups for each method. The Friedman test on $RE_{95\%}$ ranks indicates significantly different performances between the methods, both when 0 or 80 days are labeled (p-values < 0.001). At p = 0.05 and 80 labeled days, the Nemenyi test finds that SSDO is significantly better than all baselines, including kNNo. This is an important result, because *Colruyt* experts need to inspect a large number of stores on a daily basis (Section VI-B).

*b) **Q2:** Impact of the label propagation parameters:* SSDO's parameter k controls how many instances are updated by propagating the label of a single labeled instance. To assess SSDO's sensitivity to this parameter, we vary k while keeping all the other parameters fixed. Fig. 4a shows SSDO's average test set AUROC for each value of k as function of the number

TABLE I: Average test set AUROC rank ± SD for each method across all 81 window groups. SSDO ranks better than the other baselines, both without labels and after 80 labeled days.

Method	0 labeled days	80 labeled days
SSDO	**2.290 ± 1.165**	**1.889 ± 0.903**
kNNo	2.481 ± 1.123	2.630 ± 1.012
CBLOF	2.907 ± 1.412	3.123 ± 1.594
LOF	3.654 ± 1.344	3.988 ± 1.356
SSAD	5.043 ± 1.052	4.580 ± 0.954
ColruytBase	4.623 ± 1.832	4.790 ± 1.945

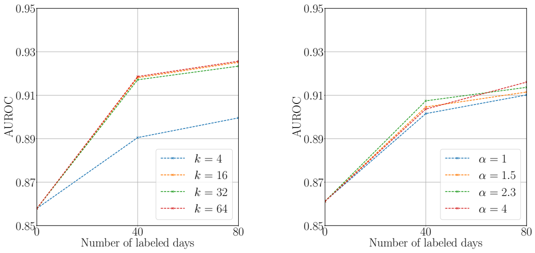
TABLE II: Average test set $RE_{95\%}$ rank ± SD for each method across all 81 window groups. SSDO ranks better than the other baselines, both without labels and after 80 labeled days.

Method	0 labeled days	80 labeled days
SSDO	**2.309 ± 1.244**	**1.802 ± 0.852**
kNNo	2.642 ± 1.240	2.852 ± 1.278
CBLOF	2.741 ± 1.438	3.037 ± 1.644
LOF	3.815 ± 1.325	4.185 ± 1.177
SSAD	4.815 ± 1.156	4.284 ± 1.021
ColruytBase	4.679 ± 1.818	4.840 ± 1.902

[6] The critical difference of the Nemenyi test with p = 0.05, 81 datasets, and 6 methods is 0.838.

of labeled days. Setting $k = 4$ results in too little propagation, and this failure to fully exploit the labeled examples degrades SSDO's performance. However, all other values of k result in nearly identical performance, regardless of the number of labeled days.

SSDO's parameter α controls the weight given to the unsupervised and label propagation components of an example's anomaly score. To assess α's impact on SSDO's performance, we vary it while keeping all the other parameters fixed. Fig. 4b shows SSDO's average test set AUROC for each value of α as function of the number of labeled days. The results indicate that α's value only has a very limited influence on SSDO's overall performance.



(a) Impact of k.

(b) Impact of α.

Fig. 4: The impact of the two label propagation parameters on the test set AUROC, averaged over all window groups: (a) shows the impact of k; (b) shows the impact of α.

*c) **Q3:** Importance of the features:* To assess the value of adding each feature category, we run SSDO with different combinations of feature categories and report $\text{RE}_{100\%}$, averaged over all window groups. SSDO with only summary statistics yields the worst average $\text{RE}_{100\%}$ of 2.84. Combining statistics with descriptive features reduces required effort to 2.44 on average. Combining statistics with shape features reduces effort to 2.35. SSDO with all features yields the lowest $\text{RE}_{100\%}$ of 2.30, a 19 percent reduction versus using only summary statistics.

VI. Deployment

Our system is currently deployed to monitor water usage in 20 stores operated by *Colruyt Group*. It is used on a daily basis to do a post-mortem analysis of the previous day's water usage in each of these stores. The system is run as web-based service that returns a ranking of potentially anomalous periods of usage. Next, we describe the system's day-to-day use, outline its architecture, provide illustrative examples of its deployed performance, and discuss lessons learned.

A. Day-to-Day System Operation

Each day at 8am, *Colruyt* engineers generate a CSV file that contains the previous day's water consumption, in five-minute increments, for each of the 20 supermarkets. These files are manually uploaded via a web-based interface and then our anomaly detection algorithm is applied to the data. The algorithm returns a global ranking over all stores for each hour of the previous day. We additionally report excess water consumption as the surplus consumption for a specific hour and store over the expected normal usage for that hour and store. The technical department, which is responsible for maintenance of the supermarkets, reviews this information each morning and plans an action according to the cost of the leakage. For a large leakage, there is a direct intervention. For a small leakage, it is added as task for the technician's next scheduled visit. Finally, the technical department gives feedback via the web-interface as to whether the detected anomalies are indeed anomalies, in order to improve the performance of SSDO.

B. System Architecture

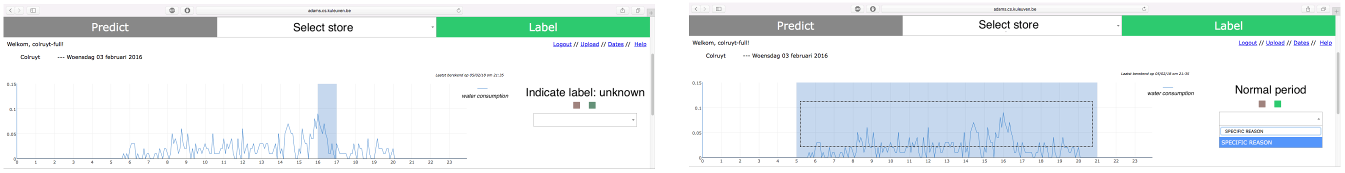
The system consists of two parts: a web-interface which is connected to a database running on a server. The end-user can access the information in the database through the web-interface, which is separated into two main views: the *predict* view and the *label* view. The *predict* view allows the user to inspect the outcome of the anomaly detection algorithm by selecting a store (or all stores) and a date range. A ranking of the days within the selected range, ordered from most to least anomalous, is returned. The user can easily give feedback by interactively selecting a part of the signal, and indicating the appropriate label, which is then automatically stored in the database. The *label* view presents the windows selected by the algorithm for feedback. The user can provide feedback on the requested windows, or select a range interactively by click-and-drag on the time series (see Fig. 5). Finally, the interface also allows for uploading new data.

The database stores the time series data, user feedback, and predictions. When adding new data, we segment the data hourly, and convert the resulting windows into the feature-vector representation, stored in the database. For each store that currently has available data in the database, we run our anomaly detection algorithm on each of the 24 window groups, and store the predictions.

C. Value of Deployment

Our approach SSDO is able to detect anomalies that the value-based alarms (ColruytBase) cannot. Hence, deploying SSDO has both reduced costs for *Colruyt* and provided an environmental benefit through minimizing water losses. Fig. 6 illustrates three examples that contrast the behavior of the two systems.

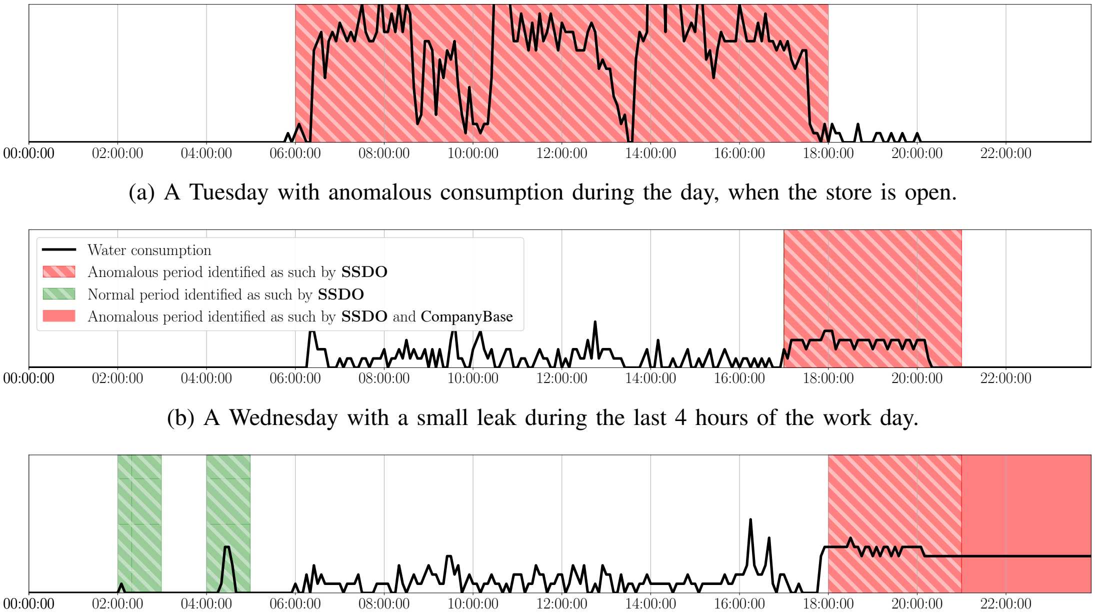
The advantages of SSDO over ColruytBase are particularly evident during the day, when the water consumption fluctuates in a stochastic manner. Fig. 6a shows an example of an anomaly detected by our algorithm during normal opening hours that was missed by the value-based alarms. Fig. 6b shows an example of a small leakage during operating hours detected by SSDO but not by ColruytBase. Small leaks are difficult to detect, particularly near the end of the day, when the activities of the in-store butchery (e.g., cleaning) increase consumption for a few hours.



(a) The system queries the user for feedback on a single window.

(b) By click-and-drag, the user labels multiple hours simultaneously.

Fig. 5: Screenshots of the web-interface's *label* view. Left: The system queries the user for feedback on one particular window. Right: The user decides to label a large part of the day as normal behavior in one go.



(a) A Tuesday with anomalous consumption during the day, when the store is open.

(b) A Wednesday with a small leak during the last 4 hours of the work day.

(c) A Thursday with a normal maintenance pattern (i.e., water softener) between 2am and 4am, and a leak starting at 6pm.

Fig. 6: Three days from different stores when SSDO correctly identified periods of anomalous water consumption (the hours shaded red) not picked up by ColruytBase. The day in plot (c) starts with an infrequent maintenance pattern between 2am and 4am (hours shaded green). SSDO correctly identifies this period as normal behavior.

Fig. 6c shows a day that starts with an infrequent, automatic maintenance pattern that results in a small amount of water usage when there is typically no consumption. SSDO correctly identifies this as normal behavior by learning from the user feedback whereas experts must hand code this domain knowledge into the ColruytBase system. At the end of the day, there is a leak that continues into the following day. SSDO identifies the leak three hours earlier than ColruytBase does.

D. Lessons Learned: Challenges and Solutions

We summarize four take-away messages. First, feature fusion is important. The behavior of the water consumption depends on various external influences, some can be inferred (e.g., opening hours) but others cannot be measured directly in a cost-efficient manner (e.g., the number of toilet visits, what meat needs to be prepared, etc.). This often results in stochastic behavior. Therefore, in Section IV-C, we use heterogeneous descriptors to characterize the water consumption time series. Second, feature construction is important. Measurements are only recorded at certain time intervals by pulse-based water flow meters. The minimum recorded water consumption is equal to a single pulse. This is a common strategy to reduce the data to transfer, but introduces quantization noise. As a result two identical usage scenarios might show different measurements because they are triggered at different times. We tackled this in Section IV-C by computing summary statistics and applying fuzzy clustering of time series to find typical shapes. Third, both anomalous and normal patterns can be recurring (e.g., certain types of leaks and maintenance patterns). On the other hand, various distinct patterns constitute normal behavior, some of which may occur less frequently than the anomalous patterns (e.g., specific maintenance operations). In Section III-B, we use user feedback to correct the assumptions of unsupervised anomaly detection. Fourth, it is impossible to obtain fully labeled data (e.g. time constraints). At most, a small amount of normal and anomalous behavior will be labeled. In Section III-C, we employ an active learning strategy to guide the user to provide feedback that is helpful to the detection algorithm, and allow the user to provide labels in a simple and efficient manner.

VII. Conclusions

This paper introduced a novel semi-supervised approach for anomaly detection. When no labeled data are available, the approach functions in an unsupervised manner by calculating an anomaly score based on a clustering step. It employs an active-learning strategy to acquire labels from the end-user. After obtaining labels, the model operates in a semi-supervised mode by using the labels to guide the clustering in the form of constraints, and then in a subsequent label-propagation step.

We discussed how to apply our approach to the real-world use case of monitoring water consumption in supermarkets. This is a challenging setting because water consumption has a complex pattern involving the combination of both stochastic behavior due to external influences, and fixed behavior. Furthermore, there are multiple distinct types of normal behavior, some of which occur less frequently than some known anomalies. We validated our approach on several years worth of data from four stores provided by the Belgian retailer *Colruyt*, and compared its performance to unsupervised and state-of-the-art semi-supervised anomaly detection techniques as well as *Colruyt*'s existing approach. Empirically, after acquiring a small number of labels, our approach outperformed the competing techniques. Currently, our approach is deployed and used by *Colruyt* in 20 stores on a daily basis.

Acknowledgment

This work was supported by the SBO grant HYMOP of the Flemish Innovation & Entrepreneurship agency (VLAIO). Jesse Davis is partially supported by the KU Leuven Research Fund (C14/17/070) and FWO-Vlaanderen (SBO-150033).

References

[1] R. B. Randall, *Vibration-based condition monitoring: industrial, aerospace and automotive applications*. John Wiley & Sons, 2011.

[2] S. W. Choi, C. K. Yoo, and I.-B. Lee, "Overall statistical monitoring of static and dynamic patterns," *Industrial & engineering chemistry research*, vol. 42, no. 1, pp. 108–117, 2003.

[3] V. Venkatasubramanian, R. Rengaswamy, S. N. Kavuri, and K. Yin, "A review of process fault detection and diagnosis: Part III: process history based methods," *Computers & Chemical Engineering*, vol. 27, no. 3, pp. 327–346, 2003.

[4] R. Langone, C. Alzate Perez, B. De Ketelaere, J. Vlasselaer, W. Meert, and J. Suykens, "LS-SVM based spectral clustering and regression for predicting maintenance of industrial machines," *Engineering Applications of Artificial Intelligence*, vol. 37, pp. 268–278, 2015.

[5] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys*, vol. 41, no. 3, pp. 1–58, 2009.

[6] European Environment Agency. (2010) Water scarcity. [Online]. Available: http://www.eea.europa.eu/themes/water/featured-articles/water-scarcity

[7] J. Spinoni, J. V. Vogt, G. Naumann, P. Barbosa, and A. Dosio, "Will drought events become more frequent and severe in europe?" *International Journal of Climatology*, vol. 38, no. 4, pp. 1718–1736, 2017.

[8] M. A. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko, "A review of novelty detection," *Signal Processing*, vol. 99, pp. 215–249, 2014.

[9] M. Gupta, J. Gao, C. C. Aggarwal, and J. Han, "Outlier detection for temporal data: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 9, pp. 2250–2267, 2014.

[10] M. Raciti, J. Cucurull, and S. Nadjm-Tehrani, "Anomaly detection in water management systems," in *Critical Infrastructure Protection*. Springer, 2012, pp. 98–119.

[11] M. Fagiani, S. Squartini, L. Gabrielli, M. Severini, and F. Piazza, "A statistical framework for automatic leakage detection in smart water and gas grids," *Energies*, vol. 9, no. 9, pp. 665–690, 2016.

[12] S. Patabendige, R. Cardell-Oliver, R. Wang, and W. Liu, "Detection and interpretation of anomalous water use for non-residential customers," *Environmental Modelling & Software*, vol. 100, pp. 291–301, 2018.

[13] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, vol. 29, no. 2. ACM, 2000, pp. 427–438.

[14] V. Jakkula and D. Cook, "Outlier detection in smart environment structured power datasets," in *Sixth International Conference on Intelligent Environments*. IEEE, 2010, pp. 29–33.

[15] H. Janetzko, F. Stoffel, S. Mittelstadt, and D. A. Keim, "Anomaly detection for visual analytics of power consumption data," *Computers & Graphics*, vol. 38, pp. 27–37, 2014.

[16] J.-S. Chou and A. S. Telaga, "Real-time detection of anomalous power consumption," *Renewable and Sustainable Energy Reviews*, vol. 33, pp. 400–411, 2014.

[17] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: identifying density-based local outliers," in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, vol. 29, no. 2. ACM, 2000, pp. 93–104.

[18] S. Papadimitriou, H. Kitagawa, P. B. Gibbons, and C. Faloutsos, "LOCI: Fast outlier detection using the local correlation integral," in *Proceedings of the 19th International Conference on Data Engineering*. IEEE, 2003, pp. 315–326.

[19] V. Hautamaki, I. Karkkainen, and P. Franti, "Outlier detection using k-nearest neighbour graph," in *Proceedings of the 17th International Conference on Pattern Recognition*, vol. 3. IEEE, 2004, pp. 430–433.

[20] J. Tang, Z. Chen, A. W.-C. Fu, and D. W. Cheung, "Enhancing effectiveness of outlier detections for low density patterns," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Berlin, Heidelberg: Springer, 2002, pp. 535–548.

[21] Z. He, X. Xu, and S. Deng, "Discovering cluster-based local outliers," *Pattern Recognition Letters*, vol. 24, no. 9–10, pp. 1641–1650, 2003.

[22] G. O. Campos, A. Zimek, J. Sander, R. J. Campello, B. Micenková, E. Schubert, I. Assent, and M. E. Houle, "On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study," *Data Mining and Knowledge Discovery*, vol. 30, no. 4, pp. 891–927, 2016.

[23] N. Görnitz, M. Kloft, K. Rieck, and U. Brefeld, "Toward supervised anomaly detection," *Journal of Artificial Intelligence Research*, vol. 46, pp. 235–262, 2013.

[24] S. Das, W. Wong, T. G. Dietterich, A. Fern, and A. Emmott, "Incorporating expert feedback into active anomaly discovery," in *Proceedings of the 16th IEEE International Conference on Data Mining*. IEEE, 2016, pp. 853–858.

[25] T. Pevný, "Loda: Lightweight on-line detector of anomalies," *Machine Learning*, vol. 102, no. 2, pp. 275–304, 2016.

[26] K. Wagstaff, C. Cardie, S. Rogers, and S. Schrödl, "Constrained k-means clustering with background knowledge," in *International Conference on Machine Learning*, vol. 1, 2001, pp. 577–584.

[27] V. S. Sheng, F. Provost, and P. G. Ipeirotis, "Get another label? improving data quality and data mining using multiple, noisy labelers," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge Discovery and Data Mining*. ACM, 2008, pp. 614–622.

[28] D. D. Lewis and W. A. Gale, "A sequential algorithm for training text classifiers," in *Proceedings of the 17th annual international ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 1994, pp. 3–12.

[29] J. G. Cleary and I. Witten, "Data compression using adaptive coding and partial string matching," *IEEE Transactions on Communications*, vol. 32, no. 4, pp. 396–402, 1984.

[30] T. Rakthanmanon, B. Campana, A. Mueen, G. Batista, B. Westover, Q. Zhu, J. Zakaria, and E. Keogh, "Searching and mining trillions of time series subsequences under dynamic time warping," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge Discovery and Data Mining*. ACM, 2012, pp. 262–270.

[31] A. F. Emmott, S. Das, T. Dietterich, A. Fern, and W.-K. Wong, "Systematic construction of anomaly detection benchmarks from real data," in *Proceedings of the ACM SIGKDD workshop on outlier detection and description*. ACM, 2013, pp. 16–21.

[32] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine learning research*, vol. 7, pp. 1–30, 2006.