Robust Hierarchical Clustering for Novelty Identification in Sensor Networks: With Applications to Industrial Systems

Sepehr Maleki, Chris Bingham

School of Engineering, University of Lincoln

Lincoln, United Kingdom

Abstract

The paper proposes a new, robust cluster-based classification technique for Novelty Identification in sensor networks that posses a high degree of correlation among data streams. During normal operation, a uniform cluster across objects (sensors) is generated that indicates the absence of novelties. Conversely, in presence of novelty, the associated sensor is clustered distinctly from the remaining sensors, thereby isolating the data stream which exhibits the novelty. It is shown how small perturbations (stemming from noise, for instance) can affect the performance of traditional clustering methods, and that the proposed variant exhibits a robustness to such influences. Moreover, the proposed method is compared with a recently reported technique, and shown that it performs 365% faster computationally. To provide an application case study, the technique is used to identify emerging fault modes in a sensor network on a sub-15MW industrial gas turbine in presence of other abrupt, but normal changes that visually might otherwise be interpreted as malfunctions.

Keywords: Novelty Detection, One-Class Classifier, Hierarchical Clustering, Artificial Intelligence, Sensor Networks, Fault Detection and Isolation.

1. Introduction

Novelty detection is regarded as the task of discovering data whose characteristics differ from that available during training or otherwise designated normal in some capacity. The practical advantages of novelty detection techniques have benefited many application areas where typically a large number of normal observations (termed positive examples) are obtained with the view to identify statistically significant anomalies in the subsequent data. Common examples include robotics [5], medical diagnostic [1, 2], failure detection in complex industrial systems [3], fraud detection [4], and sensor networks [6, 7]. In almost all cases, the underlying dynamic characteristics of the particular system provides an important insight for selection and design of the novelty detection mechanism. However, such models that represent the dynamic characteristics, are not always readily available, especially when considering interactions of networks of complex subsystems. Moreover, the designer is unlikely to be able to determine all potential scenarios that could lead to the generation of novelty even when "good" models are available. Therefore, novelty detection methods often rely on hidden features that are extracted by statistical analysis of the relevant data (see [8] for a survey of widely used techniques).

Novelty detection algorithms are generally categorised into either machine-learning-, or statistical-based techniques. Statistical techniques themselves are divided into non-parametric and parametric methods. Parametric methods model the data by assuming some underlying distribution (e.g., Normal), whereas, non-parametric methods do not make such a-priori assumptions [15]. Machine-learning techniques, on the other hand, are defined by classification tasks where the goal is to learn a model that correctly classifies an unseen object into a correct subclass of data. Machine-learning-based classifiers can be considered as either supervised or unsupervised depending on whether or not a correctly subclassed data is used for training a model. For simplicity, multi-class supervised classification problems for novelty detection are conventionally decomposed into several binary classification tasks. For such problems, a set of training samples $X = \{(\mathbf{x}_i, \theta_i) | \mathbf{x}_i \in \mathbb{R}^D, i = 1, ..., N\}$, where each sample consists of a D dimensional vector $\mathbf{x}_i$ and a corresponding class $\theta_i \in \{-1, 1\}$ are given. From this training dataset, a function $\mathcal{H}(\mathbf{x})$ that maps a given input $\mathbf{x}^\star$ to an estimate of one of the two targets is constructed.

Typically, despite the increasing availability of data, the lack of appropriate training examples remains one of the key challenges for novelty detection on complex industrial systems. Consequently, in applications where novelty is rare, but can be catastrophic, unsupervised learning schemes are often the most appropriate choice. Despite the reporting of many diverse novelty detection techniques, the resulting performance benefits are typically very application-specific [16]. For instance, robust Principal Component Analysis (PCA) [9] and its neural network equivalent, robust autoencoders [10] are perhaps the most widely used unsupervised novelty detection techniques. Whilst these have been demonstrated to achieve satisfactory novelty detection performance in some applications where a portion of multi-sensor data exhibits novelty, they often fail to correctly isolate novelty when widespread abnormality across the collective dataset is present. Hierarchical clustering is another common unsupervised technique used for structural analysis of data [11, 12, 13]. Despite outputting a graphical representation that illustrates the natural clusters within the data, the representation can be quite complex to interpret, specially when dealing with large datasets [14]. Moreover, small perturbations (such as noise) can significantly alter the cluster arrangements.
Also with data getting bigger and bigger storage limitations [17] become another challenge. This is compounded by a common requirement for novelty detection algorithms to be able to be applied on streaming data.

This paper extends and builds on the preliminary results originally reported in [6] by providing an industrial scenario that illustrates where such algorithms are needed, analysing the sensitivity and robustness of the algorithm, investigating the buffer length effect on threshold selection, scenarios where the algorithm needs to reject a false alarm along with an additional real industrial scenario, and verifying how the algorithm performs compared to other alternatives. Therefore, a robust hierarchical clustering algorithm for novelty detection in sensor networks is developed that addresses the aforementioned issues that might come up as a consequence of using classical methods. The algorithm can be applied both offline and online on streaming data and its output is readily interpreted. The notations used in this paper are given in Table 1.

Notation	Definition
N	Number of individual data streams
$\mathbf{x}$	New data sequence of dimension N
$\mathbf{S}$	Vector of N variables used for calculation of incremental variance
$\text{buffer}_{\text{len}}$	Fixed size of the buffer
buffer_i	i-th buffer of length $\text{win}_{\text{length}}$, $i = 1, 2, \ldots, N$
L	Number of data-points in win_i
$\mathbf{mean}_{\text{buffer}}$	Vector of N mean values for data-points within their respective windows
$\mathbf{mean}_{\text{pop}}$	Vector of N mean values for the whole dataset (population) so far
$\mathbf{mean}_{\text{pre}}$	Vector of N mean values for the whole data-set at the previous timestamp
$\mathbf{std}_{\text{pop}}$	Vector of N standard deviation values for the whole dataset at current time
$\mathbf{std}_{\text{score}}$	Vector of N values indicating number of standard deviations which an observation is above or below the corresponding $\mathbf{mean}_{\text{pop}}$
SE	Standard error of the data set at current time
data_{len}	Length of the dataset at current timestamp
$\oslash$	Element-wise division operator
$\sqrt[\circ]{\ }$	Element-wise square root operator
$\circ$	Schur product operator

Table 1: Notations

In the sequel, an overview of classical hierarchical clustering is initially provided in Section 2 and its shortcomings for novelty detection are discussed by use of an example. Section 3 provides the primary contribution of the paper, where the robust hierarchical clustering algorithm for novelty detection is developed. In so doing, it is seen that the algorithm exploits the correlation among sensors to reduce the computational complexity. Moreover, to further address the storage limitation issues, an updating mechanism is used that only stores the current "state" of the novelty detection procedure whilst forgetting all the previously stored states. In Section 4, a number of industrial scenarios are used to demonstrate the efficiency and advantages of the proposed methodology. Section 5, compares the algorithm with a number of available alternatives. Finally, Section 6, concludes the paper by summarising the performance of the algorithm.

2. Classical Hierarchical Clustering

Hierarchical cluster analysis is commonly used for structural analysis of data. From the similarities between the objects of a dataset, typically described by a pair-wise distance matrix, a graphical representation (conventionally a dendrogram) is generated that hierarchically clusters the most similar objects together. Contrarily, objects that are clustered furthest apart represent those that are 'most different'.

To determine which clusters should be formed or split, a measure of similarity between sets of observations is required. This is often obtained via an appropriate metric (that describes the distance between pairs of observations), and a linkage criterion which specifies the similarity of sets as a function of the pairwise distances of observations.

While hierarchical clustering does not require a pre-specified number of clusters as input, its output is more informative and structured.

Consider a set of N objects and a corresponding $N \times N$ pair-wise distance matrix, the basic process of Johnson's hierarchical clustering [18] is summarised below:

I. Each object initially is assigned an individual cluster resulting in formation of N clusters. Distances between the clusters are given by the distance among the containing objects.

II. Determine the closest cluster pair and merge them to form a single cluster; the number of clusters is therefore reduced to $N - 1$.

III. Calculate the distances between the newly formed cluster and the remaining ones. Distance measures can be obtained through various metrics and linkage methods, eg., *complete-link*, *single-link*, and *average-link* [19]. In complete-link clustering (also called the maximum method), the distance between one cluster and another cluster is considered to be the maximum distance from any member of one to any member of the other. In the single-link method (also termed the minimum method), the distance between two clusters is considered to be the minimum distance from any member of one to any member of the other. Finally, in the average-link method, the distance between two clusters is considered to be the average distance from any member of one to any member of the other.

IV. Repeat steps II and III until all items are included inside a single cluster of size N.

While the outlined procedure for hierarchical clustering outputs an informative hierarchy that gives a general sense of the data structure (in terms of proximity of measurements) and its underlying patterns, it is not capable of novelty identification except during some uncommon conditions (e.g., a considerably large outlier). Moreover, small perturbations and even different clusters aggregation methods for the same data set can produce different hierarchies and hence different partitions [20].

2.1. An Illustrative Example

Consider a pair-wise distance matrix obtained from a single set of measurements from 6 burner-tip temperature sensors $\{S_1, S_2, ..., S_6\}$ on an industrial gas turbine, given as follows. Absolute differences in temperature (°C) are used as the distance measure in this case.

$$\begin{array}{c|cccccc|}
 & S_1 & S_2 & S_3 & S_4 & S_5 & S_6 \\
S_1 & 0 & 116.48 & 154.73 & 33.22 & 50.33 & 18.65 \\
S_2 & 116.48 & 0 & 38.71 & 18.03 & 66.42 & 97.97 \\
S_3 & 154.73 & 38.71 & 0 & 121.93 & 104.64 & 136.26 \\
S_4 & 33.22 & 18.03 & 121.93 & 0 & 18.03 & 15.02 \\
S_5 & 50.33 & 66.42 & 104.64 & 18.03 & 0 & 31.87 \\
S_6 & 18.65 & 97.97 & 136.26 & 15.02 & 31.87 & 0
\end{array}$$

The task is to find the sensors that have the most (or the least) similarities in their measurements. To achieve this, the four stages of the hierarchical clustering, outlined above, is applied using the complete-link method. As can be observed from the distance matrix, the minimum pair-wise distance is the corresponding distance for (S_4 and S_6). Therefore, the pair is selected to form a cluster with a height that indicates the distance of these sensors (see Figure 1).

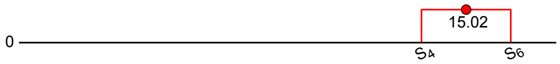


Figure 1: First cluster formed via the complete-link method.

The newly formed cluster is subsequently considered as a composite measurement and the new measurement set becomes:

$$\{S_1,\, S_2,\, S_3\ (S_4,\, S_6),\, S_5\}\ .$$

The pair-wise distance between the composite (S_4, S_6), and the remaining sensors are considered to form the distance matrix for the second stage of hierarchical clustering:

$$
\begin{array}{cc}
 & \begin{array}{ccccc} S_1 & S_2 & S_3 & (S_4, S_6) & S_5 \end{array} \\
\begin{array}{c} S_1 \\ S_2 \\ S_3 \\ (S_4, S_6) \\ S_5 \end{array} &
\begin{bmatrix}
0 & 116.48 & 154.73 & 33.22 & 50.33 \\
116.48 & 0 & 38.71 & 97.97 & 66.42 \\
154.73 & 38.71 & 0 & 136.26 & 104.64 \\
33.22 & 97.97 & 136.26 & 0 & 31.87 \\
50.33 & 66.42 & 104.64 & 31.87 & 0
\end{bmatrix}
\end{array},
$$

from which, the second cluster is formed analogously between S_5 and (S_4, S_6) (see Figure 2).

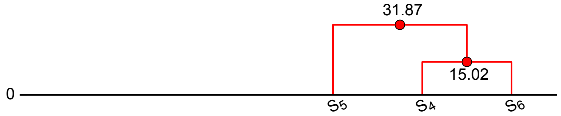


Figure 2: Second cluster formed via the complete-link method.

The process continues until only a single cluster is formed that contains all the previously formed clusters (Figure 3):

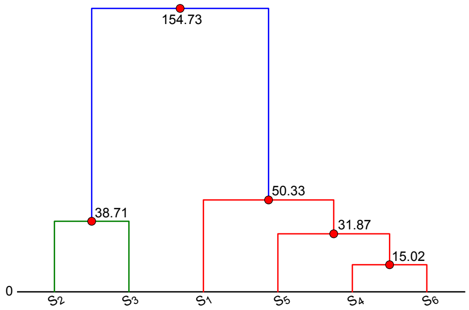


Figure 3: Final cluster generated using the complete method.

2.2. Effects of perturbations and sensor inaccuracy

Consider the distance matrix obtained for the second cluster of the previous example. The smallest pair-wise distance is calculated as 31.87 and between $\big(S_5,\ (S_4, S_6)\big)$. However, note the next minimum distance which is between $\big(S_1,\ (S_4, S_6)\big)$ and calculated as 33.22. The two inferred distances can be considered essentially identical in the presence of perturbations and sensor inaccuracy. For example, considering the burner-tip temperatures of an Industrial Gas Turbine (IGT), typical temperature measurements are around 800°C. Therefore, a very small change in the measurements can consequently change the cluster arrangement significantly resulting in inconsistency of the outcome of classical hierarchical clustering algorithm for the intended application of novelty detection.

3. Robust Hierarchical Clustering

As seen in Section 2, despite being a powerful tool for multivariate data analysis, hierarchical clustering may not be sufficiently robust for novelty detection under realistic scenarios. Moreover, the generated hierarchy is often complex and difficult to interpret leading to several stages of post-processing being required to identify the natural clusters. These motivate the need for a new clustering technique that not only inherits the advantages of hierarchical clustering whilst addressing traditional limitations. Here then, a one-class classification method is developed that classifies the data which exhibit a high degree of correlation in the same cluster while showing robustness against noise compared to the conventional hierarchical clustering method. As a result, the generated hierarchy is also much simpler to interpret without any post-processing.

In presence of noise, the technique aims to **a.** Identify novelty in the stream of data (if it exists) **b.** Identify the source of novelty. To achieve this, the next section of the paper is subdivided into two parts. Firstly, the robustness issue is addressed. Secondly, a robust one-class clustering technique for online novelty detection is developed.

3.1. Robustness

To alleviate effects of perturbations, a smoothing stage before applying the clustering method of choice (e.g., complete-link, average-link, ...) is introduced on the pair-wise distance matrix which relaxes the imposed clustering sensitivity. Consider a pair-wise distance matrix $D \in \mathbb{R}^{(N \times N)}$:

$$\begin{array}{cc}
 & \begin{array}{ccccc} S_1 & S_2 & ... & S_{N-1} & S_N \end{array} \\
\begin{array}{c} S_1 \\ S_2 \\ \vdots \\ S_{N-1} \\ S_N \end{array} &
\begin{bmatrix}
0 & d_{12} & ... & d_{1(N-1)} & d_{1N} \\
d_{21} & 0 & ... & d_{2(N-1)} & d_{2N} \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
d_{(N-1)1} & d_{(N-1)2} & ... & 0 & d_{(N-1)N} \\
d_{N1} & d_{N2} & ... & d_{N(N-1)} & 0
\end{bmatrix}
\end{array},$$

and $D^\star$ is given by:

$$\begin{array}{c|ccccc|}
 & m_1 & m_2 & \dots & m_{k-1} & m_k \\
\hline
m_1 & 0 & d^\star_{12} & \dots & d^\star_{1(k-1)} & d^\star_{1k} \\
m_2 & d^\star_{21} & 0 & \dots & d^\star_{2(k-1)} & d^\star_{2k} \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
m_{k-1} & d^\star_{(k-1)1} & d^\star_{(k-1)2} & \dots & 0 & d^\star_{(k-1)k} \\
m_k & d^\star_{k1} & d^\star_{k2} & \dots & d^\star_{k(k-1)} & 0
\end{array}.$$

3.1.1. Illustrative Example

Consider the following distance matrix $D \in \mathbb{R}^{3\times3}$ is given ($N = 3$):

$$D = \begin{array}{c|ccc|}
 & S_1 & S_2 & S_3 \\
\hline
S_1 & 0 & 116.48 & 117.01 \\
S_2 & 116.48 & 0 & 38.71 \\
S_3 & 117.01 & 38.71 & 0
\end{array}.$$

The matrix describes the pair-wise distances between three sensors $\{S_1, S_2, S_3\}$. To construct D^*:

1. Construct the set $\mathcal{D}$ whose elements are the upper (or lower) triangular block of D. That is, $\mathcal{D} = \{116.48, 117.01, 38.71\}$. Moreover, denote

members of $\mathcal{D}$ by $m_k, k \in \{1, 2, ..., \frac{3(3-1)}{2}\}$. That is $m_1 = 116.48, m_2 = 117.01, m_3 = 38.71$.

2. Compute the elements of $D*$ which are define by:

$$d^{\star}_{pq} = |m_p - m_q|, \quad p, q \in \{1, 2, ..., \frac{3(3-1)}{2}\} .$$

More precisely:

$$d^{\star}_{1,1} = |m_1 - m_1| = 0 \quad d^{\star}_{1,2} = |m_1 - m_2| = 0.53 \quad d^{\star}_{1,3} = |m_1 - m_3| = 77.77$$

$$d^{\star}_{2,1} = |m_2 - m_1| = 0.53 \quad d^{\star}_{2,2} = |m_2 - m_2| = 0 \quad d^{\star}_{2,3} = |m_2 - m_3| = 78.3$$

$$d^{\star}_{3,1} = |m_3 - m_1| = 77.77 \quad d^{\star}_{3,2} = |m_3 - m_2| = 78.3 \quad d^{\star}_{3,3} = |m_3 - m_3| = 0$$

which results in:

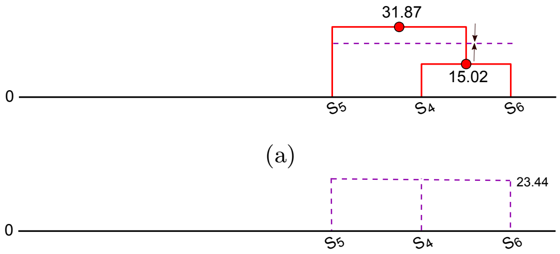
$$D^* = \begin{array}{c|ccc|} & m_1 & m_2 & m_3 \\ m_1 & 0 & 0.53 & 77.77 \\ m_2 & 0.53 & 0 & 78.3 \\ m_3 & 77.77 & 78.3 & 0 \end{array} .$$

By introducing a *tolerance threshold*, η, the elements that satisfy $d^{\star}_{pq} \leq \eta$ indicate that their corresponding pairs (d_{pq}, d_{qp}), in the original distance matrix D are susceptible to perturbations and therefore are replaced with their mean value.

Applying this iterative smoothing algorithm on the illustrative example detailed in 2.1, the two clusters of Figure 2 now merge to form a single cluster (see Figure 4). The process continues until data from all of the sensors form a single cluster.

Algorithm 1: Iterative Smoothing Algorithm.

```
while Receiving data do
    # $D \in \mathbb{R}^{(N \times N)}$ is the pair-wise distance matrix.
    $\mathbf{D}^\star$ = zeros $(N, N)$
    for $i = 1$ to $N$ do
        for $i = j$ to $N$ do
            if min $(|\mathbf{D}$ - $\mathbf{D}[i,j] \times \mathbf{I}_{N \times N}|) < \eta$ then
                # Get the index of the closest pair to $\mathbf{D}[i,j]$
                $[k,l]$ = index(min($|\mathbf{D} - \mathbf{D}[i,j] \times \mathbf{I}_{N \times N}|$))
                $\mathbf{D}^\star[i,j] \longleftarrow$ mean($\mathbf{D}[i,j], \mathbf{D}[k,l]$)
                $\mathbf{D}^\star[j,i] \longleftarrow \mathbf{D}[i,j]$
    # Generate clusters from the new distance matrix.
    return cluster $(\mathbf{D}^\star)$
```



(a)

(b)

Figure 4: (a) The two clusters generated as a result of classical hierarchical clustering. (b) A single cluster, constructed by applying the iterative smoothing algorithm.

3.2. A Robust One-Class Technique For Online Novelty Detection

Now consider N streams of highly correlated data and define a fixed-size buffer for each stream, respectively, that stores the incoming data on a First In First Out (FIFO) basis. The data stored in these buffers are referred to as samples, whereas the whole data obtained so far is the population. For each data-point that enters a buffer, a *standard score* is calculated (see Algorithm 2) which indicates how the new information has changed the sample distribution with respect to its respective population. If a novel data-point enters one of the buffers, it is expected to result in the generation of a significantly different standard score. To exploit the correlation amongst the data, these standard scores are adopted to calculate a pair-wise distance matrix. Then, classical clustering with the smoothing stage is applied across the buffers that ultimately results in the formation of a single cluster if novelties are not present in the data. Alternatively, the smoothing stage, and specifically the condition imposed by η can be viewed as a requirement that the objects of the distance matrix, D, to ultimately fall inside the same cluster unless they are significantly distant (i.e., novel) from the rest. In this latter case, a new cluster is formed by the significantly distant object. In case where all objects (sensors) exhibit abnormal characteristics, the height of the generated dendrogram will be several times greater than those of normal profiles and therefore can be thresholded to raise an alarm.

For completeness, the robust hierarchical clustering algorithm is summarised below:

I. Rolling windows of N streams of data with a predefined buffer size L are captured to read the incoming signals, i.e., a FIFO buffer.

II. Algorithm 2 is applied to data streams to generate the corresponding standard scores.

III. From the standard scores, a pair-wise distance matrix is generated.

IV. Hierarchical clustering with the pre-smoothing stage is performed on the distance matrix generated from the standard scores (Algorithm 1).

Algorithm 2: Real-time Standard Score Calculation Algorithm.

```
Initialise: data $_{\text{len}}$ = 0
            for $i = 1$ to $N$ do
                buffer $_i$ = buffer of fixed size buffer $_{\text{len}}$
            end
            $\mathbf{S} = \mathbf{zeros}\ (N)$
            $\mathbf{std}_{\text{ scores}} = \mathbf{zeros}\ (N)$
            $\mathbf{mean}_{\text{ win}} = \mathbf{zeros}\ (N)$
            $\mathbf{mean}_{\text{ global}} = \mathbf{zeros}\ (N)$
while Receiving data do
    Let: $\mathbf{x}$ = new data
    increment: data $_{\text{len}}$
    $L = \mathbf{min}(\text{buffer}_{\text{ len}}, \text{ data}_{\text{ len}})$
    Let: $\mathbf{mean}_{\text{ pre}} = \mathbf{mean}_{\text{ pop}}$
    $\mathbf{mean}_{\text{ new}} = \mathbf{mean}_{\text{ pre}} + \frac{(\mathbf{x} - \mathbf{mean}_{\text{ pre}})}{\text{data}_{\text{ len}}}$
    $\mathbf{S} = \mathbf{S} + (\mathbf{x} - \mathbf{mean}_{\text{ new}}) \circ (\mathbf{x} - \mathbf{mean}_{\text{ pre}})$
    $\mathbf{std}_{\text{pop}} = \sqrt[\circ]{\frac{\mathbf{S}}{\text{data}_{\text{len}}}}$
    $\mathbf{SE} = \frac{\mathbf{std}_{\text{pop}}}{\sqrt{L}}$
    Let: $\mathbf{mean}_{\text{ pop}} = \mathbf{mean}_{\text{ new}}$
    for $i = 1$ to $N$ do
        buffer $_i$.append ($\mathbf{x}$[i])
    $\mathbf{mean}_{\text{ buffer}} = \mathbf{Mean}\ (\text{buffer}_i)$
    $\mathbf{std}_{\text{ scores}} = |\mathbf{mean}_{\text{ buffer}} - \mathbf{mean}_{\text{ pop}}| \oslash \mathbf{SE}$
```

3.3. The Tolerance Level η

The tolerance level η plays a pivotal role in the cluster reduction process of the developed algorithm. Here, a methodology based on quantile analysis of the data and originally reported in [3], is adopted to determine a suitable η to minimise false-alarms.

Definition 1. *[21] Consider a random variable X with the probability distribution function $\mathcal{F}$ and let $0 < p < 1$. A value x_p is called a* quantile of order p *if*

$$\mathbb{P}\{X < x_p\} \leq p \leq \mathbb{P}\{X \leq x_p\} ,$$

or equivalently

$$\mathcal{F}(x_p - 0) \leq p \leq \mathcal{F}(x_p) .$$

Normalising quantiles between 0 and 100 results in a relative measure known as percentile.

In pursuit of a suitable η, Algorithm 3 is applied to a carefully selected training dataset that does not contain novelty. Distance matrices are generated and corresponding minimum pairwise distances are stored. The quantile function (inverse Cumulative Distribution Function (CDF)) is then applied on the stored distances to determine the higher observation quantiles. Then, given a desired reliability n, the n-th percentile is selected as η.

Algorithm 3: Algorithm For Selecting The Tolerance Level η.

```
Initialise: $\mathbf{T}_{\text{stack}} = [\,]$

while Receiving data do
    store(data)
    Run: Algorithm 1
    get(std $_{\text{score}}$ )
    generate (D)
    # $D_{\triangle}$ is the lower triangular part of $D$.
    $\Delta \longleftarrow$ sort(unstack($D_{\triangle}$))
    for $i = 1$ to $(N \times N) - N/2$ do
        L $\leftarrow \Delta[i+1] - \Delta[i]$
    $\mathbf{T}_{\text{stack}}$.append (min (L))
# quantile(q) computes the q-th quantile.
# n is the desired reliability level.
$\eta = \mathbf{T}_{\text{stack}}$.quantile($n$)
```

3.4. Buffer Size

The proposed algorithm utilises rolling windows (buffer) of size L to calculate the standard scores. As a parameter of the algorithm, the buffer size L used during the so-called "training" process should remain the same in the "evaluation". For completeness, the impact of various buffer sizes on the performance of the algorithm is analysed.

At each time-step that all windows roll forward to absorb the incoming measurements, a set of standard scores are generated (one score per window). Standard scores are the signed number of standard deviations by which the value of an observation or data point is above or under the mean value of what is being observed or measured thus far. The generated scores at each time interval provide information on how a single incoming measurement changes the sample, compared to the population. Hence, with a larger buffer size a greater change in the measurements is required to change the sample with respect to the population. Figure 5 shows how the window length (buffer size) affects the sensitivity of the algorithm.

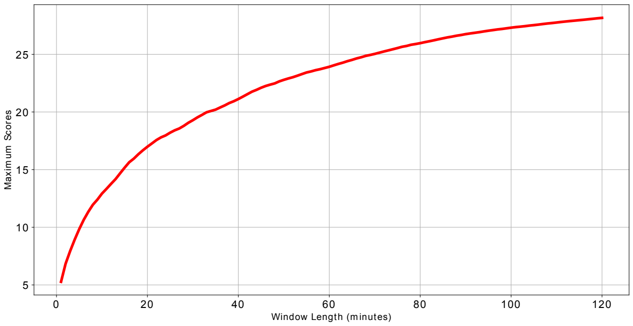
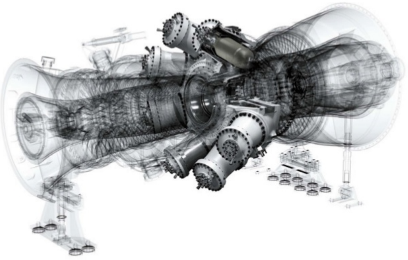


Figure 5: Effect of using various window lengths.

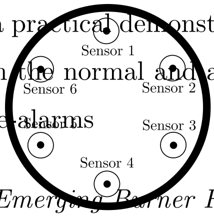
In spite of sensitivity, since the threshold is determined with the same buffer size as used later for the novelty detection procedure, ultimately, performance of the algorithm is more or less independent of the choice of the buffer size if the same parameters are used and the same sensitivity is desired.

4. Industrial Case Studies

To demonstrate the value of the proposed methodology and provide an application focus, fault detection and isolation problem for an industrial system is considered. Specifically, a set of thermocouple sensors measuring burner-tip temperatures of a Siemens IGT are chosen. All measurements are obtained from the actual operation of the engine in the field. As shown in figures 6a and 6b, burners are accommodated in 6 circumferential equidistant combustion chambers. Considering the relative proximity of the burners, it is expected to observe a high degree of correlation among the data during normal operation of the engines. Importantly, due to operational reasons (e.g., change of load), measurements can contain abrupt change-points that are not a indicative of an



(a)



(b)

Figure 6: (a) Structure of a gas turbine. (b) Annular array of burners in an IGT

impending malfunction and should be considered normal. Failure to detect an impending malfunction, can result in serious structural damages (e.g., deformation and cracking) and consequently longer maintenance down-time. Therefore, this provides a practical demonstration for how the developed classifier discriminates between the normal and abnormal operating conditions without unduly triggering false alarms

4.1. Case 1: Emerging Burner Failure

Initially, Algorithm 3 is applied and from the resulting CDF plot (see Figure 7), a threshold value of 21 corresponding to the 95% detection confidence level is chosen.

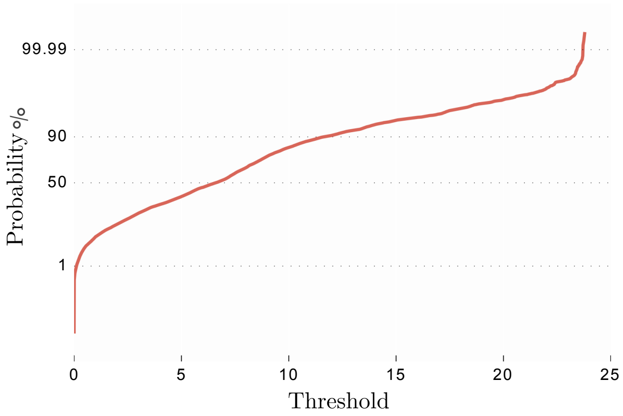


Figure 7: CDF plot for threshold determination.

Choosing a lower confidence level (e.g., 90%) results in a higher false-alarm rate. However, a higher confidence level may miss some of the statistical indications of an impending failure and detect the emerging fault comparatively later [3]. The first practical scenario considers operation of an IGT where one of the sensors monitoring burner-tip temperatures exhibits abnormalities (see Figure 8). Despite the malfunction, the engine was kept operational afterwards, incurring further sub-sequential damage.

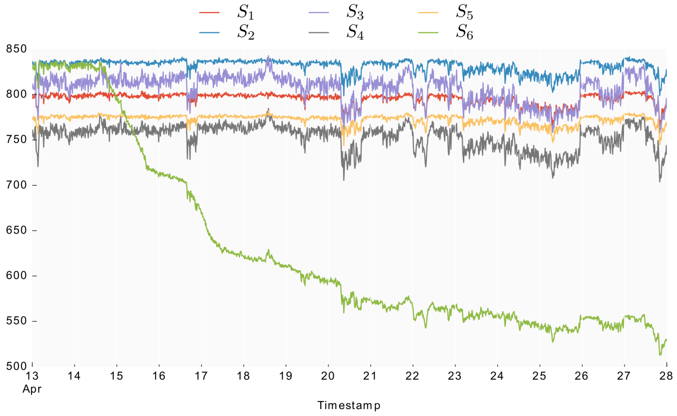


Figure 8: Thermocouple measurements for the 6 sensors (degrees °C). Sensor 6 exhibits abnormalities.

To demonstrate the advantages of the robust hierarchical clustering, 30 days of data from the 6 thermocouples are collected prior the to the malfunction day. By applying the algorithm, subsequent clusters are generated for each time window of 1 hour. Figure 9 provides an example of such generated clusters that indicate a normal operation (i.e., absence of any novelties).

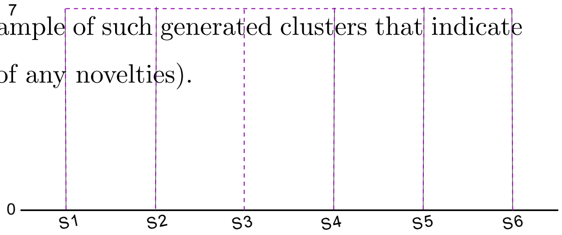


Figure 9: Dendrograms generated before the malfunction.

However, on the day the malfunction occurs (see Figure 10), as the signs of the malfunction emerge, cluster formation also changes to indicate presence of novelty. Specifically, the sensor that exhibits the novelty (i.e., $S6$) is clustered separately with respect to rest of the sensors, as can be seen from Figure 11.

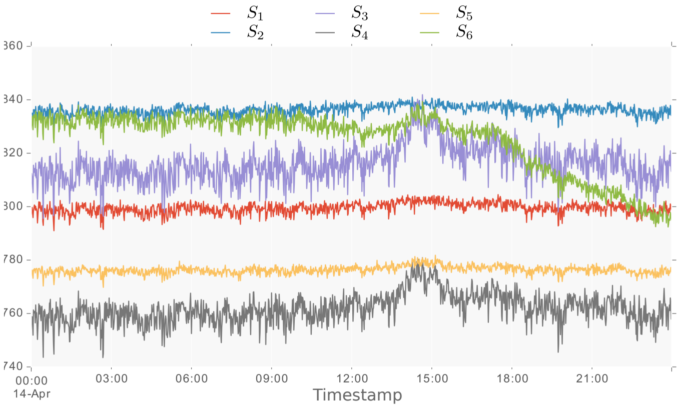


Figure 10: Thermocouple measurements for the 6 sensors (degrees °C) on the day of malfunction.

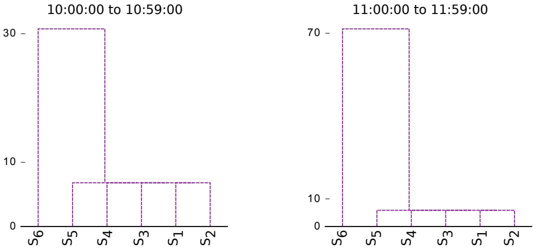


Figure 11: Generated clusters as the result of the emerging malfunction.

Moreover, on the day of malfunction (Figure 10), the data trend changes multiple times for each sensor. Specifically, a sudden change measured by S_3 and S_4 is seen at about 15:00 while S_6 indicates the emergence of a malfunction. This change is not flagged by the robust hierarchical clustering algorithm as a result of a correctly chosen tolerance level, η, thereby successfully identifying the emerging malfunction without raising any false-alarms.

4.2. Case 2: Rejecting False Alarms

As previously described, an important feature of novelty detection schemes is minimising the number of false-alarms whilst also being significantly sensitive enough to detect emerging failures in a timely manner. A practical example is now given using another set of burner-tip temperature measurements taken over a period of 1 day (see Figure 12). During the period depicted in Figure 12 it is known that no failure was present on the unit although the measurement data visually contains abnormal characteristics that would typically be perceived/detected as indicative of a failure or an emerging fault. Specifically, it can be seen that multiple periods of transient behaviour of the unit, and hence the temperature characteristics. However, these expected changes are due to an intentional change of load. While, individually, any of these periods could be considered as indicating a fuel system fault to the respective burner, the robust hierarchical clustering algorithm does not identify a fault in this instance since there is a collective behavioural change exhibited by all sensors. An alarm is therefore not raised and the engine continues to operate and a dendrogram

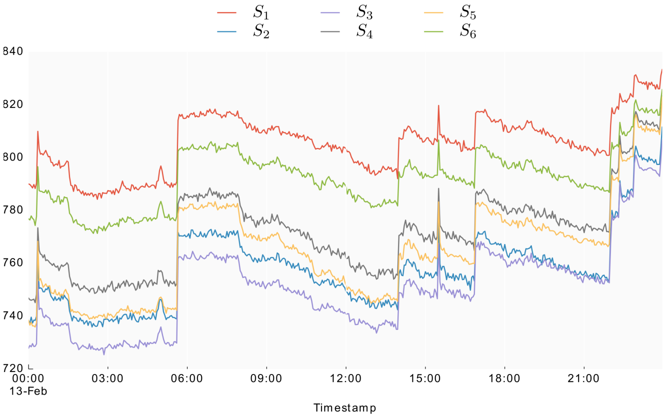


Figure 12: Burner temperatures (degrees °C) over a period of 1 day.

similar to that of Figure 9 is generated by the robust hierarchical clustering algorithm during the period of 1 day indicating no malfunctioning has occurred during the test period.

5. Comparison With Alternative Methods

In this section, the proposed method is compared with two other techniques that are commonly employed to detect novel characteristics in sensor networks.

5.1. Hierarchical Cluster Analysis

Classical Hierarchical cluster analysis is one of the common techniques used for classification in sensor networks (see e.g., [23] for a comprehensive review).
Performing either single-link, average-link, or complete-link hierarchical clustering analysis for the case study represented in Section 4.1, results in inconsistent clusters, some of which are shown in Figure 13 :

Figure 13: Dendrograms generated via hierarchical cluster analysis.

As can be observed from Figure 13 the dendrograms indicate a large deviation of sensors S_5 (a) and S_3 (b) while in fact S_6 is the sensor whose measurements indicate a malfunctioning. A comparison of Figure 13 (c) and the corresponding time slot shown in Figure 11 reveals that using the robust hierarchical clustering algorithm, the malfunctioning could have been detected and isolated successfully while the classical hierarchical cluster analysis methods fail to detect at all.

5.2. Changepoint Detection

Recently an online 2-D changepoint detection algorithm for sensor-based fault detection, has been proposed [3]. The methodology consists of **i)** a *differential detector* which analyses characteristics across datasets at a particular instant, and **ii)** a *standard detector* which when combined can detect anomalies through identification of meaningful changepoints. While the changepoint detection algorithm successfully detects and isolates failures, the computational effort is much greater (hence takes longer to execute) than that of the robust hierarchical clustering algorithm. To verify this, both algorithms are applied under the same conditions on a number of industrial scenarios and the runtime for each algorithm is measured (see Table 2 for the average runtime comparison).

	Changepoint Detection	Robust Hierarchical Clustering
Average Runtime	5.1 (s)	1.4 (s)

Table 2: Robust hierarchical clustering vs. Changepoint Detection runtime

In addition to a longer runtime, the changepoint detection algorithm requires two thresholds to correctly determine the time and location of the failure. Therefore, two stages of data processing are required to determine the thresholds, namely, one stage to determine the threshold for the standard detector and the other to determine that for the differential detector. However, the robust hierarchical clustering algorithm requires only one stage of data processing to determine the noise tolerance level. To a first approximation, the set-up time is therefore essentially halved.

6. Conclusions

A one-class clustering algorithm, termed robust hierarchical clustering, for novelty identification in highly correlated datasets (e.g., those obtained from a sensor network) is developed. As a result, a uniform cluster of all objects (sensors) is generated where novelties do not exist. However, in presence of a novelty, the formation of clusters change such that the object (sensor) that contains the novelty is clustered separately. Efficacy of the classifier is examined in a number of actual industrial case studies where it is shown how the classifier discriminates between those trend changes that are due to operational reasons and those that are indicative of an emerging malfunction. Such algorithms play an important role in industries where the limited workforce cannot meet the demands of daily monitoring of a large number of systems. Compared to other proposed techniques (e.g., [3]), the algorithm executes 364% faster and requires only a single threshold to be determined a-priori. Whereas for the technique proposed in [3], two thresholds are required. Despite the outlined advantages, the algorithm needs to be trained on “clean” data where no malfunctions are present. Although the training is a one-time process, for some industries it might not be immediately feasible. This is left for future work, where unsupervised training of the algorithm will be investigated.

Acknowledgement

The authors would like to thank Siemens Industrial Turbomachinery, Lincoln, U.K., for providing access to real-time data to support the research outcomes.

References

[1] L. Clifton, D. Clifton., P. Watkinson, and L. Tarassenko, *Identification of patient deterioration in vital-sign data using one-class support vector machines*,

Federated Conference on Computer Science and Information Systems, IEEE, 2011, pp. 125–131.

[2] A. Duraj, *Outlier detection in medical data using linguistic summaries*, IEEE International Conference on INnovations in Intelligent SysTems and Applications (INISTA), Gdynia, 2017, pp. 385-390.

[3] S. Maleki, C. Bingham, and Y. Zhang, *Development and realisation of changepoint analysis for the detection of emerging faults on industrial systems*, IEEE Trans. Ind. Informat., vol. 12, 2016, pp. 1180–1187.

[4] D. Olszewski, *Fraud detection using self-organizing map visualizing the user profiles*, Knowledge-Based Systems, Vol. 70, 2014, pp. 324–334.

[5] P. Ross, A. English, D. Ball, B. Upcroft, and P. Corke, *Online novelty-based visual obstacle detection for field robotics*, IEEE International Conference on Robotics and Automation (ICRA), 2015, pp. 3935–3940.

[6] S. Maleki, and C. Bingham, *A one-class Clustering technique for Novelty Detection and Isolation in sensor networks*, IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA), Annecy, 2017, pp. 1-6.

[7] Y. Zhang, N. Meratnia, and P. Havinga, *Outlier detection techniques for wireless sensor networks: a survey*, IEEE. Commun. Surv. Tutor., Vol. 12, 2010, pp. 159–170.

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

[9] E. J. Candès, and X. Li, and Y. Ma, and J. Wright, *Robust Principal Component Analysis?*, J. ACM, Vol. 58, No. 3., 2011, pp. 11:1–11:37.

[10] C. Zhou, and R. C. Paffernorth, *Anomaly Detection with Robust Deep Autoencoders*, KDD, 2017, pp. 665–674.

[11] C. Yin, and A. Kareem, *Computation of failure probability via hierarchical clustering*, Structural Safety, Vol. 61, 2016, pp. 67-77.

[12] C. W. Hu, and S. M. Kornblau, and J. H. Slater, and A. A. Qutub, *Progeny clustering: A method to identify biological phenotypes*, Scientific reports, 5, 2015.

[13] R. Darkins, and E. J. Cooke, and Z. Ghahramani, and P. D. W. Kirk, and D. L. Wild, and R. S. Savage, *Accelerating Bayesian Hierarchical Clustering of Time Series Data with a Randomised Algorithm*, Plos One, Vol. 08, No. 4, 2013, pp. 1–9.

[14] A. Rehn, and A. Possemiers, and J. Holdsworth, *Efficient hierarchical clustering for single-dimensional data using CUDA*, Proceedings of the Australasian Computer Science Week Multiconference, ACM, 2018, pp. 14:1–14:10.

[15] M. Markou, and S. Singh, *Novelty Detection: A Review;Part 1: Statistical Approaches*, Signal Process., Vol. 83, 2003, pp. 2481–2497.

[16] R. M. Bowen, *Online Novelty Detection System: One-Class Classification of Systemic Operation*, Thesis, Rochester Institute of Technology, 2015.

[17] J. Aspen, D. Goldenberg, and Y. R. Yang, *On the computational complexity of sensor network localization*, Algorithmic Aspects of Wireless Sensor Networks: First International Workshop, Vol. 3121, 2004, pp. 32–44.

[18] S. C. Johnson, *Hierarchical clustering schemes*, Psychometrika, vol. 32, 1967, pp. 241–254.

[19] C. F. Olson, *Parallel algorithms for hierarchical clustering*, Parallel Computing, Vol. 21, 1995, pp.1313–1325.

[20] L. Sousa, and J. Gama, *The application of hierarchical clustering algorithms for recognition using biometrics of the hand*, IJAERS, vol. 1, Issue-7, 2014, pp. 14–24.

[21] M. Ahsanullah, V.B. Nevzorov, and M. Shakil, *An Introduction to Order Statistics*, Atlantis Studies in Probability and Statistics, Atlantis Press, 2013, pp. 26.

[22] S. Maleki, P. Rapisarda, and E. Rogers, *Failure identification for linear repetitive processes*, Mult. Syst. & Sig. Proc., Vol. 26, 2015, pp. 1037–1059.

[23] S. Tyagi and N. Kumar, *A systematic review on clustering and routing techniques based upon LEACH protocol for wireless sensor networks*, J. Netw. Comput. Appl., vol. 36, 2013, pp. 623–645.

[24] B. P. Welford, *Note on a method for calculating corrected sums of squares and products*, Technometrics, vol. 4, 1962, pp. 419–420.

[25] D. E. Knuth, *Art of Computer Programming*, vol. 2, 3rd ed., Addison-Wesley, 1997, pp. 232.