Anomaly Detection in Streaming Nonstationary Temporal Data

Abstract

This article proposes a framework that provides early detection of anomalous series within a large collection of non-stationary streaming time series data. We define an anomaly as an observation that is very unlikely given the recent distribution of a given system. The proposed framework first calculates a boundary for the system's typical behavior using extreme value theory. Then a sliding window is used to test for anomalous series within a newly arrived collection of series. The model uses time series features as inputs, and a density-based comparison to detect any significant changes in the distribution of the features. Using various synthetic and real world datasets, we demonstrate the wide applicability and usefulness of our proposed framework. We show that the proposed algorithm can work well in the presence of noisy non-stationarity data within multiple classes of time series. This framework is implemented in the open source R package *oddstream*. R code and data are available in the supplementary materials.

Keywords: Concept drift; Extreme value theory; Feature-based time series analysis; Kernel-based density estimation; Multivariate time series; Outlier detection.

1 Introduction

Anomaly detection in streaming temporal data has become an important research topic due to its wide range of possible applications, such as the detection of extreme weather conditions, intruders on secured premises, gas and oil leakages, illegal pipeline tapping, power cable faults, and water contamination. The rapid detection of these critical events is vital in order to protect valuable lives and/or assets. Furthermore, since these applications spend the majority of their operational life in a "typical" state, and the associated data is obtained with the help of millions of sensors, manual monitoring is ineffective and time consuming, as well as highly unlikely to be able to capture all violations (Lavin & Ahmad 2015). Thus, the development of powerful new automated methods for the early detection of anomalies in streaming signals is very timely, with far-reaching benefits.

This paper makes three fundamental contributions to anomaly detection in streaming non-stationary environments. First, we propose a framework that provides early detection of anomalies within a large collection of streaming time series data. We show that the proposed algorithm works well even in the presence of noisy signals and multimodal distributions. Second, we propose an approach for dealing with non-stationary environments (also known as "concept drift" in the machine learning literature). We reduce the collection of time series to a 2-dimensional feature space, and then apply a bivariate two-sample nonparametric test to detect any significant change in the feature distribution. The asymptotic normality of the test allows us to bypass computationally intensive re-sampling methods when computing critical values. Third, we use various datasets to demonstrate the wide applicability and usefulness of our proposed framework to several application domains.

Fiber optic sensing technology can be used to detect unusual, critical events such as power cable faults (Jiang & Sui 2009), electrical short circuits (Krohn et al. 2000), gas or oil pipeline leakages (Yoon et al. 2011, Nikles 2009), intruders to secured premises (Nikles 2009), etc. For example, a sensor cable may be attached to a fence or buried along a facility's perimeter in soil or concrete, and can detect intrusion attacks such as climbing or cutting a fence, or walking, running or crawling along a facility's perimeter (Catalano et al. 2014). A light signal pulsated through the cable is easily disturbed by changes in the physical environment, such as the temperature, strain, or pressure. Thus, changes in the intensity, phase, wavelength or transit time of light in the

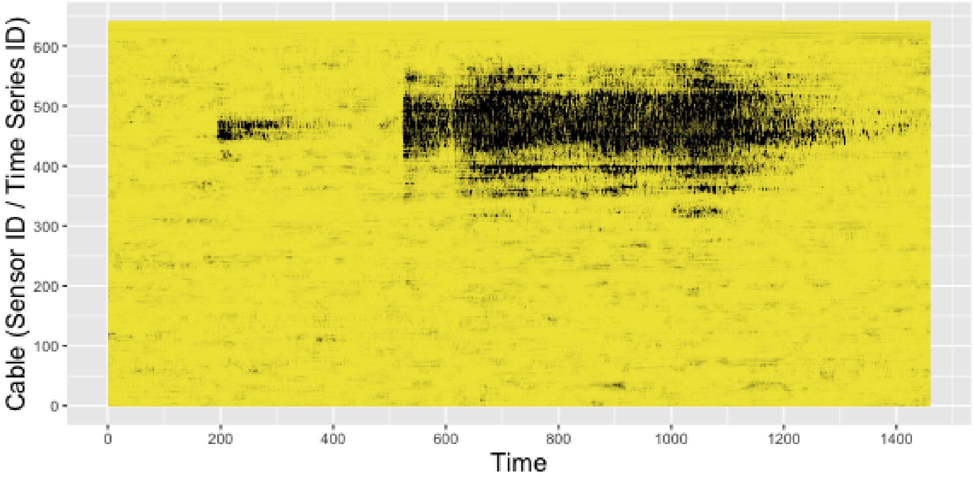

Figure 1: Multivariate time series plot of a dataset obtained using a fiber optic cable. Axis ‘Cable’ represents individual points of the sensor cable. There are 640 time series each with 1459 time points. Yellow corresponds to low values and black to high values. The black region near the upper end point of the cable (around 350 to 500) indicates the presence of an anomalous event (e.g., intrusion attack, gas pipeline leak, etc.) that has taken place during the 500–1300 time period.

fiber may indicate intrusions. Similarly, sensor cables can monitor temperature profiles along gas and oil pipelines, allowing the detection of leakages (Krohn et al. 2000). Each point of the cable acts as a sensor and generates a time series. Figure 1 shows the multivariate time series obtained using a fiber optic cable. (As the dataset contains commercially sensitive information, the actual application is not given here).

Our aim in this work is to identify the locations of unusual critical events as soon as possible. We propose an algorithm which has the ability to (a) deal with streaming data; (b) assist in the early detection of anomalies; (c) deal with large amounts of data efficiently; (d) deal with non-stationary data distributions; and (e) deal with data which may have multimodal distributions.

Section 2 presents the background work on anomaly detection for temporal data, and the use of EVT in anomaly detection. Section 3 describes the new framework for the detection of anomalies in streaming data. It also proposes a way of handling non-stationary environments. Some simulations illustrating the method are presented in Section 4. An application of the proposed framework is given in Section 5. Section 6 concludes the paper.

2 Background

2.1 Types of anomalies in temporal data

The problems of anomaly detection for temporal data are threefold: (a) the detection of contextual anomalies within a given series; (b) the detection of anomalous sub-sequences within a given series; and (c) the detection of anomalous series within a collection of series (Gupta et al. 2014).

Contextual anomalies within a given time series are single observations that are surprisingly large or small, independent of the neighboring observations. Figure 2(a) provides an example. This is a well-known problem and has been addressed by many researchers in data science (Hayes & Capretz 2015). Burridge & Taylor (2006) called these "additive outliers" and proposed an algorithm for their detection using EVT.

In contrast, when considering the detection of anomalous subsequences within a given time series, the primary focus is not on individual observations, but on subsequences that are significantly different from the rest of the sequence. An example is given in Figure 2(b). Both these problems of detecting anomalous subsequences or additive outliers can be addressed either as univariate (Bilen & Huzurbazar 2002) or multivariate problems (Riani et al. 2009, Galeano et al. 2006, Peña & Prieto 2001). The algorithm proposed by Schwarz (2008) using EVT is also capable of detecting both types of outliers, and is derived from the work of Burridge & Taylor (2006).

The final setting, the detection of anomalous series within a collection of series, is the primary focus of this paper. Figure 2(c) provides an example of this scenario. Very little attention has been paid to this problem relative to the other two problem settings. An exception is Hyndman et al. (2015) who proposed a method using principal component analysis applied to time series features, together with highest density regions and α-hulls, to identify unusual time series in a large collection of time series. The recent work of Wilkinson (2018) also has the capability to address problems of this nature.

2.2 Streaming data challenges

Approaches to the problem of anomaly detection for temporal data can be divided into two main scenarios: (1) batch processing and (2) data streams (Faria et al. 2016, Luts et al. 2014). With

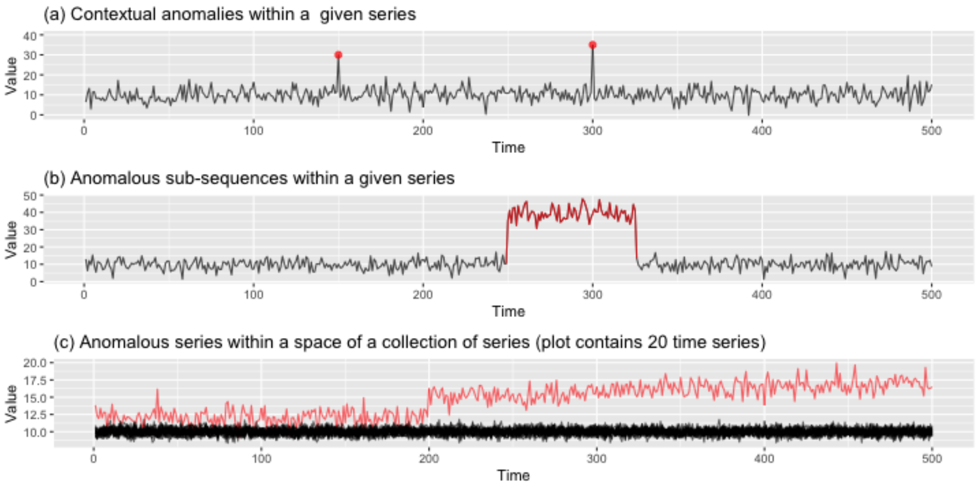

Figure 2: Different types of anomalies in temporal data. In each plot anomalies are represented by red color and black color is corresponding to the typical behavior

batch processing, as in Hyndman et al. (2015) and Wilkinson (2018), it is assumed that the entire data set is available prior to the analysis, and the aim is to detect all of the anomalies present.

The streaming data scenario poses many additional challenges, due to its complex nature and the way that the data evolve over time. Challenges include the large volume and high velocity of streaming data, the presence of very noisy signals, and nonstationary data distributions (or "concept drift"). The latter makes it difficult to distinguish between new "typical" behaviors and anomalous events. Addressing this issue requires the detecting algorithm to be able to learn from and adapt to the changing conditions. These challenges have made it difficult for the existing batch scenario approaches to provide early detection of anomalies in the streaming data context (Faria et al. 2016).

2.3 Extreme value theory for anomaly detection

Our proposed framework is based on extreme value theory (EVT), a branch of probability theory that relates to the statistical behavior of extreme order statistics (Galambos et al. 2013).

Let $\boldsymbol{X} = \{x_1, x_2, \ldots, x_m\}$ be a sequence of independent and identically distributed random variables with cumulative distribution function (CDF) F and density function $f = F'$. Let $X_{\max} = \max(\boldsymbol{X})$ and $x_i \in \Re$. The distribution of $X_{\max}$ can be investigated by taking several random samples of size m from a given distribution, recording the maximum of each sample, and constructing a density plot of the maxima. A similar approach can be used for the distribution of the minimum. Figure 3 (reproduced from (Hugueny 2013), p.87) shows the empirical distributions of minima and maxima for the standard Gaussian distribution (left), and of maxima for the standard exponential distribution (right) for series of sizes m. Each density plot is based on 10^6 data points. Consider the case of $m = 1$, where we observe only one data point from f in each trial. The corresponding density plot approximates the generative distribution f, as the maximum of a singleton set $\{x\}$ is simply x. However, the density plots for maxima move to the right as m increases, implying that the expected location of the sample maximum on the x-axis increases as more data are observed from f. Let H^+ denote the distribution function of $X_{\max}$. This is termed the *extreme value distribution* (EVD), as it describes the expected location of the maximum of a sample of size m generated from f (Clifton et al. 2011). The Fisher-Tippett Theorem (Fisher & Tippett 1928), which is the basis of classical EVT, explains the possibilities for this H^+.

The following expression of the theorem has been adapted from Theorem 3.2.3 of Embrechts et al. (2013), p.121; the notation has been changed for consistency.

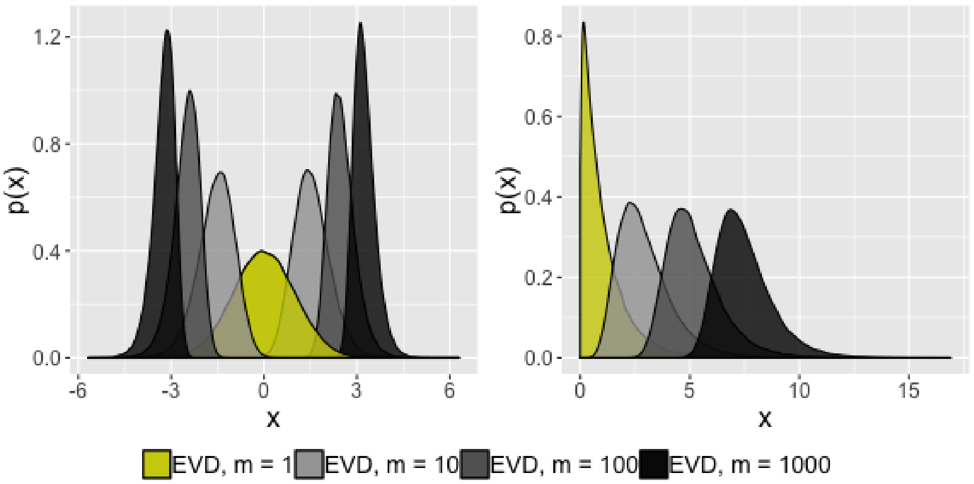

Figure 3: Empirical distributions of 10^6 minima and maxima for the standard Gaussian distribution (left), and of maxima for the standard exponential distribution (right). (Reproduced from Hugueny, 2013, p.87.)

Theorem 1 (Fisher-Tippett theorem, limit laws for maxima)

If there exists a centering constant $d_m (\in \Re)$ and a normalizing constant $c_m (> 0)$, and some non-degenerate distribution function H^+ ('+' refers to the distribution of maxima) such that $c_m^{-1}(X_{max} - d_m) \xrightarrow{d} H^+$, then H^+ belongs to one of the three distribution function types: Fréchet $\Phi^+_\alpha(x)$, Weibull $\Psi^+_\alpha(x)$ or Gumbel $\Lambda^+(x)$.

Embrechts et al. (2013) discuss some properties that assist in deciding the maximum domain of attraction (MDA) of X. If f has a truncated tail, such as the uniform or beta distribution, then it is in the MDA of the Weibull distribution. If f has an infinite tail that obeys the power law, then it is in the MDA of the Fréchet distribution. Examples include Pareto, F, Cauchy and log-gamma distributions. On the other hand, if f has an exponentially decaying tail such as the exponential, gamma, normal or log-Normal distributions, then it is in the MDA of the Gumbel distribution. Interested readers are referred to the work of Embrechts et al. (2013) for a detailed discussion of the characterization of the three classes: Fréchet, Weibull and Gumbel.

2.3.1 Existing work for anomaly detection based on extreme value theory

The literature to date has mostly defined anomalies in terms of either distance or density. When anomalies are defined in terms of distance, one would expect to see relatively large separations between typical data and the anomalies. Burridge & Taylor (2006), Schwarz (2008) and Wilkinson (2018) provide a few examples of this approach where observations with large nearest neighbor distances are defined as anomalies. Within this framework, the 'spacing theorem' (Schwarz 2008) in EVT has been used in the model building process. In contrast, defining an anomaly in terms of the density of the observations means that an anomaly is an observation that has a very low chance of occurrence. The work of Perron & Rodríguez (2003), on which the method of Burridge & Taylor (2006) was based, mentioned the possibility of using extreme value theory and non-parametric estimates of tail behavior, but did not provide any detailed discussion. Sundaram et al. (2009), Clifton et al. (2011) and Hugueny (2013) provide a few examples where EVT has been used to find observations that have extreme densities. The main focus of these methods was on defining a threshold for the density of the data points such that it distinguishes between anomalies and typical observations.

It can be seen from Theorem 1 that the EVD is parameterized implicitly by m, the size of the sample from which the extrema is taken. Thus, different values of m can yield different EVDs (Figure 3). Clifton et al. (2011) proposed a numerical method for selecting a threshold for identifying anomalous points when $m \geq 1$. In their "Ψ transform method", Clifton et al. (2011) define the "most extreme" of a set of m samples $\boldsymbol{X} = \{x_1, x_2, \ldots, x_m\}$, distributed according to pdf $f(x)$, as the most improbable with respect to the distribution; i.e., $\arg\min_{x \in X}[f(x)]$.

3 Methodology

This section proposes a new framework for anomaly detection in multivariate streaming time series based on the Ψ-transformation method proposed by Clifton et al. (2011). The proposed framework involves: (1) building a model of the typical behavior of a given system; and (2) testing newly arrived data against the model of typical behavior. These two phases represent the *off-line* (Algorithm 1) and *on-line* (Algorithm 2) phases (Faria et al. 2016) of the framework, respectively. Our proposed method is intended to overcome two limitations of the proposals of Hyndman et al. (2015) and Wilkinson (2018).

First, the method proposed by Hyndman et al. (2015) identifies the most unusual time series within a large collection of time series, whether or not any of them are truly anomalous. However, in our applications, an alarm should be triggered only in the presence of an anomalous event. Defining a boundary of typical behavior and monitoring new data points that land outside that boundary allows us to overcome this limitation as it now triggers an alarm only in the presence of an observation that lands outside the anomalous boundary.

Second, the "HDoutliers" method proposed by Wilkinson (2018) relies on the assumption that the nearest-neighbor distances of anomalous points will be significantly higher than those between typical data points. However, some applications do not exhibit large gaps between typical observations and anomalies. Instead, the anomalies deviate from the majority, or the region of typical data, gradually, without introducing a large distance between typical and anomalous observations. This is the case, for example, where the time series are highly dependent.

Consider a temperature-sensing fiber optic cable attached to a gas pipeline for the detection of gas leakages. The escape of pressurized gas changes the temperature not only at the point of the leak, but also at neighboring points, with a gradually decaying magnitude. Consequently, the observed time series will be highly dependent, with multiple anomalous points that deviate gradually from the typical behavior, without introducing a large distance between the anomalies and the typical observations.

Figure 4 illustrates this point, with panel (c) showing a large collection of time series obtained via independent sensors. For each series, we compute a vector of features which are then reduced to two principal components, plotted in panel (a). (The process of generating a feature space from a collection of time series is discussed in Algorithm 1). The two isolated points shown in black correspond to two anomalous series, and have relatively large nearest-neighbor distances compared to the typical observations shown in yellow. These large nearest-neighbor gaps allow the HDoutliers method to identify the two points as anomalies. In contrast, panel (b) represents a feature space that corresponds to a collection of time series obtained via sensors that are dependent. The corresponding multiple parallel time series plot is given in panel (d). In the example on the right, Figure 4b, the anomalous points are not widely separated from the typical points in the feature space. As the HDoutliers algorithm identifies anomalies only using the nearest neighbor distances, and there is no substantial difference between the anomalous points and the typical points, it would fail to detect these anomalous points. However, with respect to density we can see a clear separation between the anomalous points (corresponding to the low density region) and the typical points (which correspond to higher density regions) (Figure 4b). Therefore, density based approaches are more appropriate for us to choose a suitable anomalous threshold on the feature space.

Thus, we assume that anomalies have very low density values compared to those of typical points. To determine the appropriate anomalous density threshold, we use EVT taking account of the number of observations in order to properly control the probability of false positives (Clifton et al. 2011).

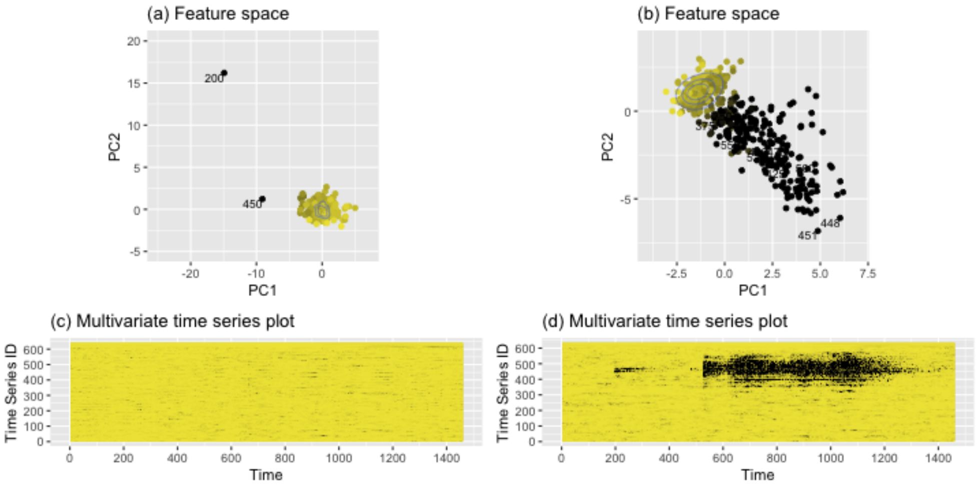

Figure 4: Left panel corresponding to a collection of time series obtained via independent sensors. Right panel corresponding to a collection of time series obtained via sensors that are not independent to one another. Black: high values, Yellow: low values. Black dots/lines/shapes are corresponding to anomalous event.

Our proposed method requires a representative dataset of the system's typical behavior. Since, by definition, anomalies are rare in comparison to a system's typical behavior, the majority of the available data must represent the given system's typical behavior. It is not necessary to have representative samples of all possible types of typical behaviors of a given system in order for the proposed algorithm to perform well. The principal idea is to have a warm-up dataset from which to obtain starting values of the parameters of the decision model.

3.1 Algorithm of the proposed framework for streaming data

Algorithm 1 (Off-line phase: Building a model of the typical behavior)

Input: D_{norm}, a collection of m time series (which can be of either equal or different lengths) that are generated under the typical system behavior.

Output: t^*, anomalous threshold.

1. Extract k features (similar to Fulcher (2012) and Hyndman et al. (2015)) from each time series in D_{norm}. This produces an $m \times k$ feature matrix, M. Each row of M corresponds to a time series and each column of M corresponds to a feature type. This feature-based representation of time series has many advantages. In this work our features have ergodic properties and are intended to measure attributes associated with non-stationarity of the time series (Kang et al. 2018). Therefore our proposed framework is well-suited for a large diverse set of time series. Further, a feature based representation of time series allows us to compare time series of different lengths and/or starting points, as we transform time series of any length or starting point into a vector of features of fixed size. It also reduces the dimension of the original multivariate time series problem via features that encapsulate the dynamic properties of the individual time series. Of the 14 features ($k = 14$) used in this work, eight (mean, variance, changing variance in the remainder (*lumpiness*), level shift using a rolling window (*lshift*), variance change (*vchange*), strength of linearity (*linearity*), strength of curvature (*curvature*), and strength of spikiness (*spikiness*)) were selected from Hyndman et al. (2015). Following Fulcher (2012), the remaining five features were defined as follows: the burstiness of the time series (Fano factor; *BurstinessFF*), minimum, maximum, the ratio of the interquartile mean to the arithmetic mean (*rmeaniqmean*), the moment, and the ratio of the means of the data that are below and above the global mean

(*highlowmu*). Figure 5 provides a feature-based representation of the time series of Figure 1.

2. Since different operations produce features over different ranges, normalize the columns of the resulting $m \times k$ feature matrix, M. Let M^* represent the resulting $m \times k$ feature matrix.

3. Apply principal component analysis to the feature matrix M^*.

4. Define a two-dimensional space using the first two principal components (PC) from step 3 (similar to Hyndman et al. (2015) and Kang et al. (2017)). Hereafter, the resulting two-dimensional PC space is referred to as the *2D PC space*. This *2D PC space* now contains m instances. Each instance on this *2D PC space* corresponds to a time series in D_{norm}. We selected only the first two PCs to maximize our chances of obtaining insights via visualization (Kang et al. 2017).

5. Estimate the probability density of this *2D PC space* using kernel density estimation with a bivariate Gaussian kernel (similar to Luca, Karsmakers, Cuppens, Croonenborghs, Van de Vel, Ceulemans, Lagae, Van Huffel & Vanrumste (2014); Cuppens et al. (2014)). Let $\hat{f}_2$ denote the estimated probability density function.

6. Draw a large number N of extremes (as defined in Clifton et al. (2011)) from $\hat{f}_2$, and form an empirical distribution of their densities in the Ψ-transform space, where the Ψ-transform of the extrema $\mathbf{x}$ is defined as

$$\Psi[f_2(\mathbf{x})] = \begin{cases} (-2ln(f_2(\mathbf{x})) - 2ln(2\pi))^{1/2}, & f_2(\mathbf{x}) < (2\pi)^{-1} \\ 0, & f_2(\mathbf{x}) \geq (2\pi)^{-1}. \end{cases}$$

   The number of instances of which we consider the extremes is m, i.e. the number of time series in the original collection D_{norm}.

7. Fit a Gumbel distribution to the resulting $\Psi[f_2(\mathbf{x})]$ values (Clifton et al. 2011, Hugueny 2013). The Gumbel parameter values are obtained via maximum likelihood estimation.

8. Determine the anomalous threshold using the corresponding univariate CDF, F_2^e in the transformed Ψ-space and thereby define a contour t^* in the *2D PC space* that describes

where the most extreme of the m typical samples generated from f_2 will lie, to some level of probability (e.g., 0.999) (Farrar & Worden 2012).

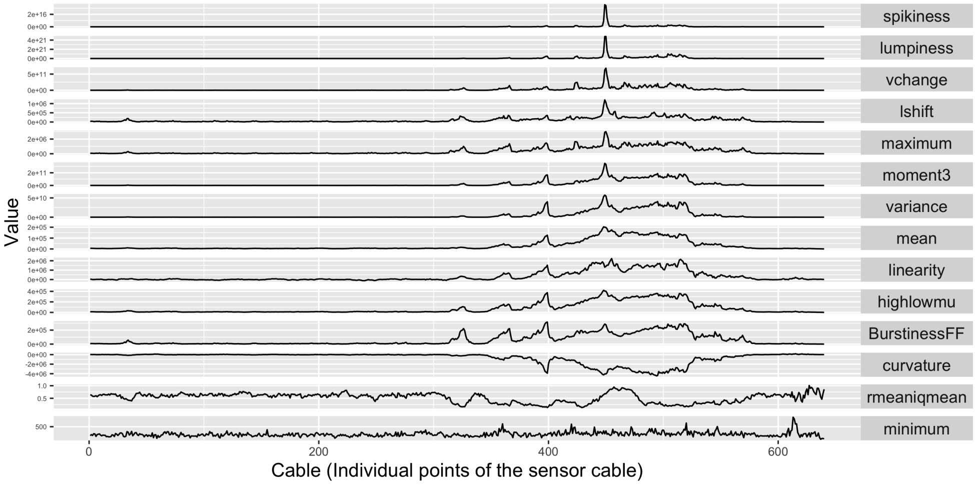

Figure 5: Feature based representation of the time series in Figure 1. There are 640 time series (m = 640). Each plot is corresponding to a feature type extracted from the 640 time series (k=14). Almost all the features have captured the unusual event near the right end point of the cable (around 350 to 550).

As recommended by Jin & Agrawal (2007), a sliding window model is used to handle the streaming data context. Given w and t, which represent the length of the sliding window and the current time point, respectively, our aim is now to identify time series that are anomalous relative to the system's typical behavior. The sliding window keeps moving forward with the current time point, maintaining its fixed window length w. As a result, the model ignores all data that were received before time $t-w$. Furthermore, each data element expires after exactly w time steps.

Algorithm 2 (On-line phase: Testing newly-arrived data)

Input: $W[t-w,t]$, the current sliding window with m time series. t^*, anomalous threshold from Algorithm 1.

Output: A vector of indices of the anomalous series within the time window $W[t-w,t]$

1. Extract k features (the features defined in step 1 of Algorithm 1) from each of the m time series in $W[t-w,t]$. This produces an $m \times k$ feature matrix M_{test}.

2. Project this new feature matrix, M_{test}, on to the same the *2D PC space* of the typical data that was built using the time series in D_{norm}. Let $\boldsymbol{Y} = y_1, y_2, \ldots, y_m$ represent data points that are obtained by projecting M_{test} on this *2D PC space*.

3. Calculate the probability density values of $\boldsymbol{Y}$ with respect to $\hat{f}_2$ in step 5 of Algorithm 1.

4. Find any y_j that satisfies $\hat{f}_2(y_j) < t^*$, where $j = 1, 2, \ldots, m$, and mark the corresponding time series (if any) as anomalous within the time window $W[t - w, t]$.

5. Repeat steps 1 - 4 of the on-line phase for every new time window that is generated by the current time point, t.

3.2 Handling non-stationary environments

The distribution of the typical behavior of a given system can change over time due to many reasons such as sensor drift, cyclic variations, seasonal changes, lack of maintenance as sensors are deployed in harsh, unattended environments, etc. (Moshtaghi et al. 2014, O'Reilly et al. 2014). In such situations, current behavior might not be sufficiently representative of future behavior (Chandola et al. 2009). Therefore it is important that our algorithm is adaptive and robust against these changes of the typical behavior over time. Cuppens et al. (2014) highlight the importance of this and mention it as a possible extension of their proposed algorithm.

In the statistics literature, this is known as non-stationarity, and it can occur in many different forms. According to O'Reilly et al. (2014), if a system has a stationary data distribution, the model from which to identify anomalies only needs to be constructed once. However in an environment with a non-stationary data distribution, it is necessary to regularly update the model in order to account for changes in the data distribution. In the econometrics literature, these non-stationary environments are sometimes classified as either "structural breaks" or "time-varying" evolutionary change (Rapach & Strauss 2008). In the machine learning literature, this phenomenon is known as "concept drift", and Gama et al. (2014) and Faria et al. (2016) describe four classes: sudden, incremental, gradual and reoccurring.

According to Gama et al. (2013), there are two approaches that can be used to adapt models in order to deal with nonstationary data distributions: blind and informed. Under the blind approach, the decision model is updated at regular time intervals without considering whether a change has really occurred or not, as in Zhang et al. (2010). This is done under the assumption that the data distribution is non-stationary (O'Reilly et al. 2014). In contrast, the informed approach updates the decision model only if a change in the data distribution is detected (Faria et al. 2016).

Under this approach the goal is to identify a time at which the data distribution changes enough to justify a model update and thereby reduce the computational complexity of the algorithm. In O'Reilly et al. (2014) these two approaches are termed 'constant update' and 'detect and retrain', respectively. According to Rodríguez & Kuncheva (2008), the former strategy is useful with gradual changes while the latter is useful with abrupt changes. The informed approach proposed by Zhang et al. (2010), updates the model of the typical behavior only when an outlier or boundary point is detected, under the assumption that they can make a significant impact on the previous model of typical behavior. However, an outlier or boundary point may not always cause a significant change in the data distribution. Moshtaghi et al. (2014) declare a change in the typical behavior when the number of consecutive anomalies detected by the algorithm exceeds a predefined threshold. Since this involves a user defined threshold, it is highly subjective and does not involve a valid probabilistic interpretation.

Following the definition of Dries & Rückert (2009), we propose an informed approach for early detection of non-stationarity that uses statistical distance measures to measure the distance between the distribution of the *2D PC space* generated from the collection of typical time series in which the latest model is defined and that generated from the typical series in the current test window. This allows us to detect whether there is any significant difference between the latest typical behavior and the new typical behavior. In an occurrence of a significant change in the data distribution, an update to the model is done using the more recent data under the assumption that data are temporally correlated, with correlation increasing as temporal distance decreases (O'Reilly et al. 2014).

Algorithm 3 (Detection of non-stationarity)

Input: w, length of the moving window. D_{t_0}, collection of m time series of length w that are generated under the latest typical behavior of a given system in which the current decision model is defined. W, test stream.

Output: A vector of indices of the anomalous series in each window

1. Estimate f_{t_0}, the probability density of the *2D PC space* defined by D_{t_0}, using kernel density estimation with a bivariate Gaussian kernel.

2. Let $W[t-w,t]$ be the current test window with m time series of length w. Extract k features

(the same features as were defined in step 1 of Algorithm 1) from each of these m time series in $W[t-w,t]$. This produces an $m \times k$ feature matrix, M_{test}.

3. Project M_{test}, onto the *2D PC space* of D_{t_0}. Let $\boldsymbol{Y}_t$ represent the newly projected data points on the *2D PC space* that correspond to $W[t-w,t]$.

4. Identify the data points on the *2D PC space* that correspond to the typical series in $W[t-w,t]$, using the anomalous threshold (output of Algorithm 1) defined using D_{t_0}. Let $\boldsymbol{Y}_{t_{\text{norm}}}(\subseteq \boldsymbol{Y}_t)\}$ represent the set of data points in *2D PC space* that correspond to the typical series of $W[t-w,t]$, and $W[t-w,t]_{norm}(\subseteq W[t-w,t])$ be the corresponding set of typical time series in $W[t-w,t]$.

5. Let p be the proportion of anomalies detected in $W[t-w,t]$. If $p < p^*$, where $p^* > 0.5$, go to step (a); otherwise, go to step (b). In the examples given in this manuscript, p^* is set to 0.5, assuming the simple ‘majority rule’. However, the user also has the option of selecting a cutoff point other than the default 0.5 in order to maximize the accuracy or incorporate misclassification costs.

a. Estimate f_{t_t}, the probability density function of $\boldsymbol{Y}_{t_{\text{norm}}}$, using kernel density estimation with a bivariate Gaussian kernel. Let $\hat{f}_{t_t}$ denote the estimated probability density function.

b. Estimate f_{t_t}, the probability density function of $\boldsymbol{Y}_t$, using kernel density estimation with a bivariate Gaussian kernel. Let $\hat{f}_{t_t}$ denote the estimated probability density function. In the case of a ‘sudden’ change, all (or most) of the points in $\boldsymbol{Y}_t$ may lie outside the anomalous boundary, defined by D_{t_0}. As a result, all (or most) of those points in $\boldsymbol{Y}_t$ will be marked as anomalies, meaning that the majority (> 0.5) is now represented by the detected anomalies. This could indicate the start of a new typical behavior. Thus, it is recommended in this situation that the decision model be updated using all of the series in the current window (instead of only the typical series detected, which now represent the minority), thereby allowing the model to adapt to the changing environment automatically. This situation is elaborated further using the synthetic datasets given in Figures 7, 8 and 9 in Section 4.2.

6. Using a suitable distance measure (e.g., the Kullback-Leibler distance, the Hellinger distance, the total variation distance, or the Jensen-Shannon distance), test the null hypothesis

$H_0 : f_{t_0} = f_{t_t}$. Since the distributions of these distance measures are unknown, bootstrap methods can be used to determine critical points for the test (Anderson et al. 1994). However, these computationally intensive re-sampling methods may prevent changes in distributions from being detected quickly, which is a fundamental requirement of most of the applications of our streaming data analysis. Therefore, following Duong et al. (2012), we test the null hypothesis $H_0 : f_{t_0} = f_{t_t}$ here by using the squared discrepancy measure $T = \int [f_{t_0}(x) - f_{t_t}(x)]^2 dx$, which was proposed by Anderson et al. (1994). Since the test statistic based on the integrated squared distance between two kernel based density estimates of the *2D PC space* is asymptotically normal under the null hypothesis, it allows us to bypass the computationally intensive calculations that are used by the usual re-sampling techniques for computing the critical quantiles of the null distribution.

7. If H_0 is rejected and $p < p^*$, D_{t_0} is set to $W[t-w,t]_{norm}$. If H_0 is rejected and $p > p^*$, D_{t_0} is set to $W[t-w,t]$.

8. Repeat steps 1–7 for every new time window that is generated by the current time point t.

4 Experiments

The effectiveness of the proposed frameworks for anomaly detection in the streaming data context is first evaluated using synthetic data (these datasets are available online in supplemental materials). When generating these synthetic datasets, care has been taken to imitate situations such as applications with multimodal typical classes, different patterns of non-stationarity, and noisy signals. However, we acknowledge that the set of examples that we have used for this discussion is relatively limited, meaning that these examples should be viewed only as simple illustrations of the proposed algorithms. We hope that the set of examples will grow over time as the performances of the proposed algorithms are investigated further.

We also performed an experimental evaluation of the accuracy of our proposed framework. All the experiments (Figure 6–10) were evaluated using common measures for binary classification such as accuracy, false positive (FP) rate and false negative (FN) rate. According to Hossin & Sulaiman (2015), these measures are not enough to measure the performance of the binary classification tasks on imbalanced datasets. Since our example datasets are highly imbalanced and are negatively dependent (i.e., containing many more typical points than anomalous points), we also recorded two additional measures which are recommended for imbalanced binary classification problems: optimized precision (OP) which remains relatively stable even in the presence of large imbalances in the data (Ranawana & Palade 2006), and positive predictive value (PPV) which measures the probability of a positively predicted pattern actual being positive (outlier). Very low PPV values can be observed for certain rolling windows in Figure 6(d)–10(d), as those windows are free from true positives (anomalous events) and that lead the PPV value to become zero for the corresponding moving windows.

4.1 Detection of anomalies in the streaming data scenario

Our leading example shown in Figure 6(a) aims to demonstrate the application of Algorithms 1 and 2. In this example it is assumed that the typical behavior of the given system has a stationary data distribution and does not change over time. In other words it is assumed that the training set is drawn from a stationary data distribution and the testing stream will also be drawn from the same distribution. Therefore the dataset is generated using a Gaussian mixture of two components with different means but equal variance such that the *2D PC space* generated by the collection of series consists of a bi-modal typical class throughout the entire period. We make the anomaly detection process more challenging by generating these time series with noisy signals. The corresponding side view of the dataset is given in Figure 6(b), and demonstrates both the nature of the noisy signals and the progress and structure of the anomalous event in the 400–1000 time period. Due to the assumption of stationarity, the anomalous threshold was set only once at $F_2^e = 0.999$ using $W[1,150]$. The anomalies detected in window $W[151,300]$ are marked at $t = 300$ in Figure 6(c), then the sliding window is moved one step forward to test for anomalies in $W[152,301]$. This process is repeated for every new time window generated by sliding the window one step forward. Over time, the grid in Figure 6(c) is filled gradually from left to right with the output produced by each sliding window.

Since the anomalous event in this dataset is placed at $t = 400$, ideally we would expect Algorithm 1 and 2 to detect it when the sliding window reaches $W[250,400]$. In Figure 6(c), the anomalies detected are marked in black. As expected, Algorithms 1 and 2 were able to detect the anomalous event right from the beginning; that is, as soon as the moving window reaches

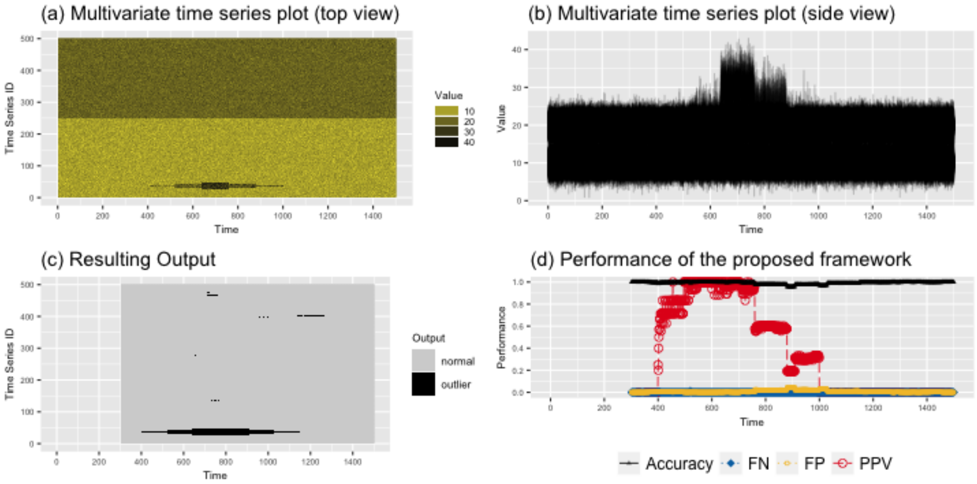

Figure 6: Multimodal typical classes but no non-stationarity. Sliding window length = 150 time points. To initiate the algorithm, $W[1,150]$ is considered as a representative sample of the typical behavior. a) Multivariate time series plot of the collection of time series ($m = 300$). The upper half of the figure (dark yellow) corresponds to one typical class, while the lower half of the figure (bright yellow) corresponds to the other typical class. b) Multivariate time series plot (side view of panel a)). c) The output produced by the sliding window approach. The anomalous threshold was set at $F_2^e = 0.999$. d) Performance of the proposed framework (without any adjustments to non-stationary environments). Overall optimized precision is 0.9904. Minimum accuracy is 0.956 (at $t = 887$). Maximum FP rate is 0.044 (at $t = 887$). Maximum FN rate is 0.014 (at $t = 520$).

$W[250,400]$. However, even though the anomalous event actually ends at $t = 1000$, as seen in Figure 6(a), the resulting output in Figure 6(c) shows that it generates an alarm until $t = 1150$. This is due to the use of a moving window of length 150, which means that the sliding window covers at least part of the anomalous event until it reaches $W[1000,1149]$. Thus, the proposed algorithm generates an alarm until it reaches a window that is completely free of the anomalous event; in this case, it stops generating an alarm once it reaches $W[1001,1151]$. This behavior of the proposed algorithm increases the FP rate immediately after the end of any anomalous event. However, in applications such as intrusion attacks to secured premises, gas/oil pipeline leakages, etc., there is no harm in generating an alarm immediately after an anomalous event ends, as this helps to capture the attention of the people who are responsible for taking the necessary action.

A sensor cable attached to a security fence for detecting intruders is one plausible application that could give rise to this type of dataset. For example, if one half of the fence is exposed to sea wind and the other half is protected by trees and buildings, this will give rise to two typical behaviors for the two halves of the same cable, as the environmental behavior can have an impact on the internal structure of the sensor cable. Similar behavior can be expected from a fiber optic cable laid along a stream for detecting water contamination. The movement of the water can have an impact on the internal structure of the sensor cable, thereby giving rise to a collection of series with multimodal typical classes at different locations along the sensor cable. For all the examples discussed under Section 4, the average accuracy is calculated by taking the ratio of the number of correctly classified series to the total number of series of each moving window generated by the current time point. As can be seen from Figure 6(d), our algorithm shows a 0.992 accuracy level on average for this dataset (Optimized precision is 0.9904), while maintaining low FP (0.0076 on average) and FN (0.000 on average) rates.

One-class support vector machine (OCSVM) is a commonly used method in anomaly detection research (Ma & Perkins 2003, Mahadevan & Shah 2009, Rajasegarar et al. 2010). Raskutti & Kowalczyk (2004) and Zhuang & Dai (2006) have proposed improved versions of OCSVM for imbalanced data where the minority class (abnormal class) is specifically targeted in the classification. However if minorities are difficult or expensive to obtained and defined OCSVM for imbalanced data is not among the best candidates for anomaly detection due to unavailability of enough instances from the abnormal class to properly train an OCSVM. Further, Luca, Karsmakers & Vanrumste (2014) highlight some limitations with OCSVM when more than one data point is observed that involves multiple hypothesis testing. Since our method does not have a direct competitor, we compared our results with HDoutliers algorithm. In each test phase HDoutliers algorithm was applied to the high dimensional space generated by the 14 features introduced in step 1 of Algorithm 1. For this dataset in Figure 6 (a) it gives a 0.988 accuracy level on average. The reported OP of 0.5356 is much lower than that of our method (Figure 6).

4.2 Anomaly detection with non-stationary environments

We now investigate the performances of Algorithm 3 together with Algorithms 1 and 2 using four synthetic datasets. Following Gama et al. (2014), these synthetic datasets are generated such that they exhibit the four different types of non-stationarity: *sudden* (a sudden switch from one distribution to another), *gradual* (trying to move to the new distribution gradually while going back and forth between the previous distribution and the new distribution for some time), *reoccurring* (a previously seen distribution reoccurs after some time) and *incremental* (there are many, slowly-changing intermediate distributions in between the previous distribution and the new distribution). The corresponding graphical representations of these four cases are given in Figures 7, 8, 9 and 10, respectively. In Figure 7(a), the anomalous event is placed in the 150th to 170th series over the time period from $t = 450$ to $t = 475$. In Figure 8(a), the anomalous event is placed in the 150th to 170th series over the time period from $t = 850$ to $t = 875$. In the remaining cases (Figures 9 and 10), the anomalous event is placed in the 150th to 170th series over the time period from $t = 825$ to $t = 875$. In all of these cases, non-stationary behavior starts to occur from $t = 300$.

In the first three cases, namely *sudden* (Figure 7), *gradual* (Figure 8 and *reoccurring* (Figure 9), when the sliding window reaches the $t = 300$ time point (i.e., $W[201,300]$), the decision model declares almost all points in that window as anomalies. As a result, p, the proportion of outliers detected in $W[201,300]$, exceeds the user-defined threshold $p*$ (set here to 0.5, based on the simple 'majority rule'). Following step 5(b) of Algorithm 3, the decision model is now

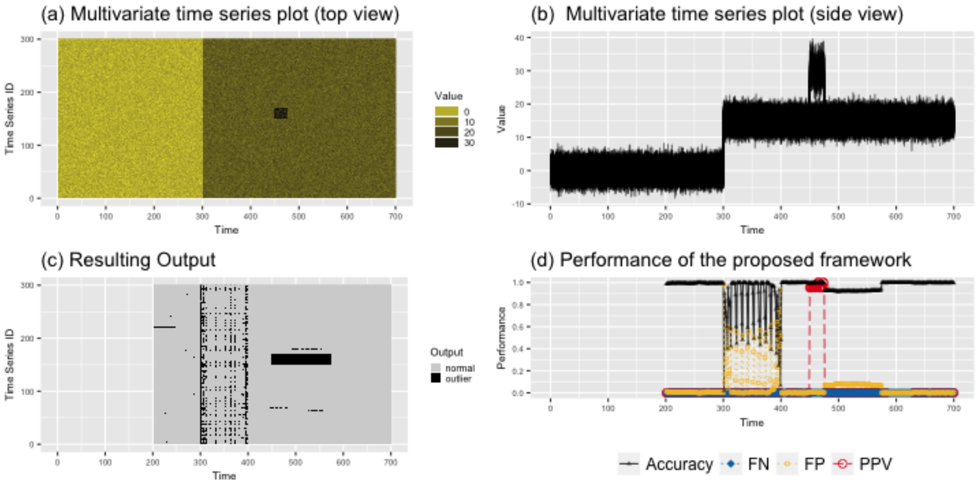

Figure 7: 'Sudden' non-stationarity. a) Multivariate time series plot of the collection of time series ($m = 300$). 'Sudden' non-stationarity starting from t =300. b) Multivariate time series plot (side view of panel a)). c) The output produced by the sliding window approach. In the test phase the anomalous threshold is updated for nonstationary behavior according to Algorithm 3 d) Performance of the proposed framework. Overall optimized precision is 0.9234. Minimum accuracy is 0.0167 (at $t = 301$). Maximum FP rate is 0.983 (at $t = 301$). Maximum FN rate is 0.0033 (at $t = 450$).

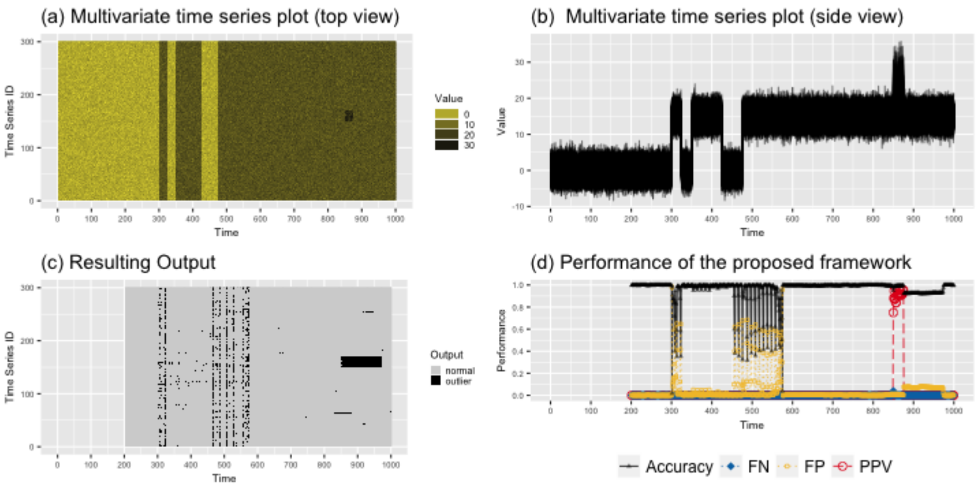

Figure 8: ’Gradual’ non-stationarity. a) Multivariate time series plot of the collection of time series ($m = 300$). ’Gradual’ non-stationarity starting from t =300. b) Multivariate time series plot (side view of panel a)). c) The output produced by the sliding window approach. In the test phase the anomalous threshold is updated for nonstationary behavior according to Algorithm 3 d) Performance of the proposed framework. Overall optimized precision is 0.9601. Minimum accuracy is 0.0167 (at $t = 301$). Maximum FP rate is 0.983 (at $t = 301$). Maximum FN rate is 0.04 (at $t = 850$).

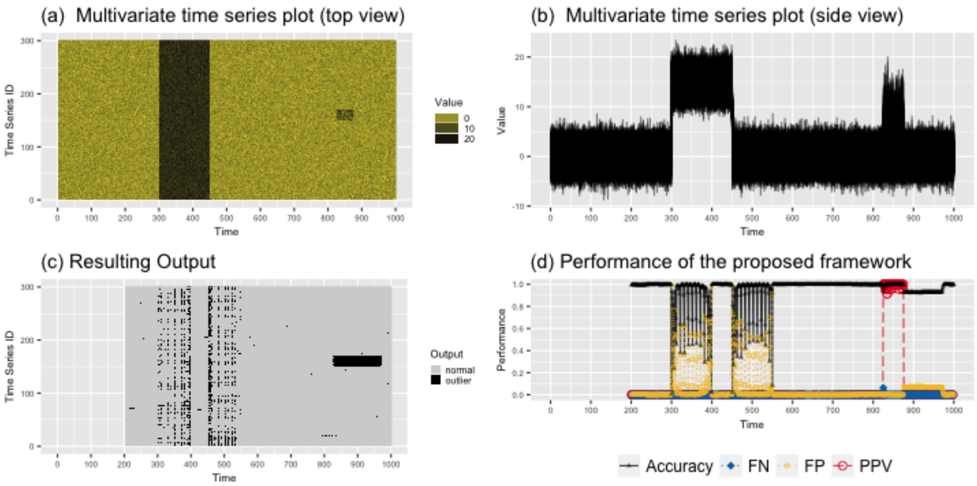

Figure 9: ‘Reoccuring’ type non-stationarity. a) Multivariate time series plot of the collection of time series ($m = 300$). ‘Reoccuring’ type non-stationarity starting from t =300. b) Multivariate time series plot (side view of panel a)). c) The output produced by the sliding window approach. In the test phase the anomalous threshold is updated for nonstationary behavior according to Algorithm 3 d) Performance of the proposed framework. Overall optimized precision is 0.9426. Minimum accuracy is 0.0067 (at $t = 300$). Maximum FP rate is 0.993 (at $t = 300$). Maximum FN rate is 0.0633 (at $t = 825$).

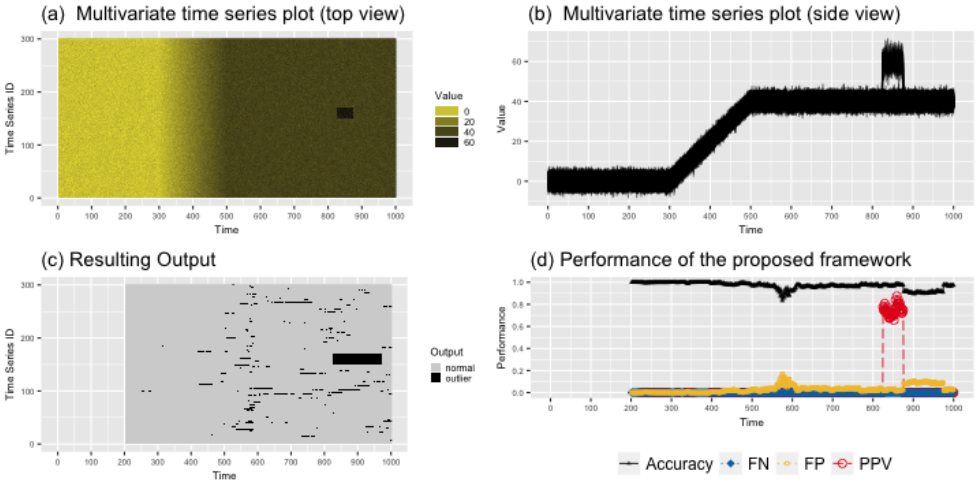

Figure 10: ’Incremental’ non-stationarity.a) Multivariate time series plot of the collection of time series ($m = 300$). ’Incremental’ non-stationarity starting from t =300. b) Multivariate time series plot (side view of panel a)). c) The output produced by the sliding window approach. In the test phase the anomalous threshold is updated for nonstationary behavior according to Algorithm 3 d) Performance of the proposed framework. Overall optimized precision is 0.953. Minimum accuracy is 0.83 (at $t = 576$). Maximum FP rate is 0.17 (at $t = 576$). Maximum FN rate is 0 (at $t = 201$).

updated using all of the series in that window, rather than just the detected ‘typical’ series which now represent the minority. This step allows the decision model to adjust to the new typical behavior if it continues to exist for a given period of time. As can be seen in plots (c) and (d) of Figures 7, 8 and 9, the decision model initially declares almost all of the series as anomalies when the non-stationarity starts to occur, but ceases to claim them as anomalies once the new pattern is established and continues to exist. After the decision model has adapted fully to the new distribution, it again starts to produce results with a high level of accuracy, while maintaining low levels of FP and FN rates.

In contrast, none of the sliding windows in our analysis of the dataset given in Figure 10(a) declare more than half of the series to be outliers. Thus, the model updating process is done based on step 5(a) of Algorithm 3 using only the typical series detected for each window. As can be seen in Figure 10(d), our proposed framework (Algorithms 1, 2, 3), shows an average level of accuracy of 0.969 (overall optimized precision 0.953) for the entire period, while maintaining low FP (0.031 on average) and FN (0.000 on average) rates during the time period under consideration.

Figure 11 illustrates the change in distribution over time via the p-value of the hypothesis test

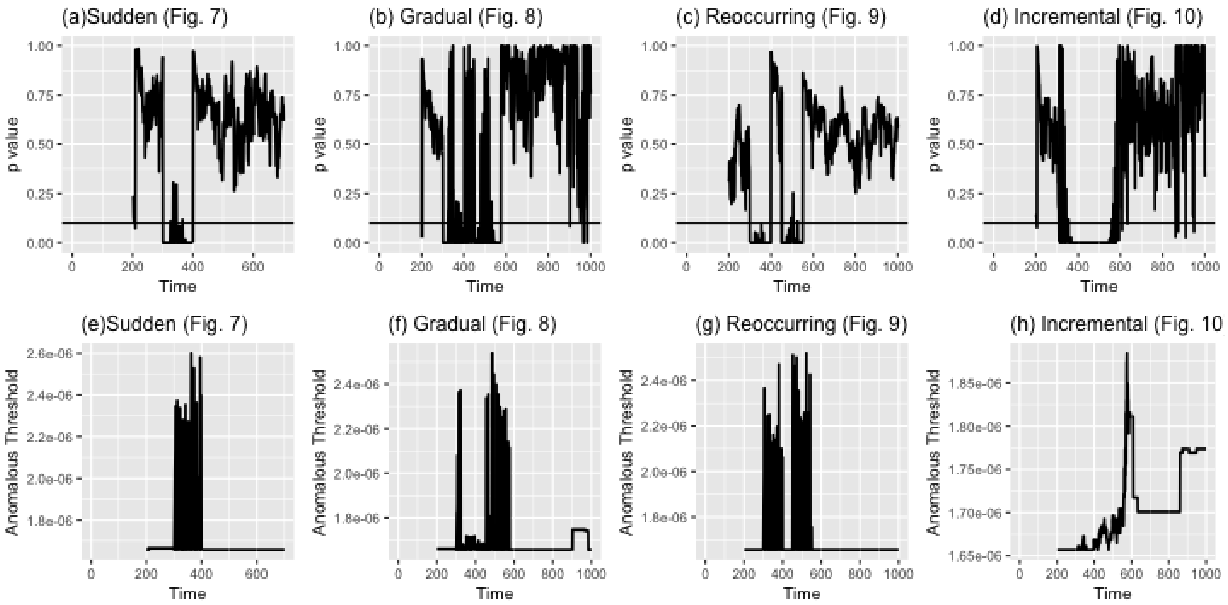

Figure 11: Detection of non-stationarity. Top panel: P value for the hypothesis test $H_0 : f_{t_0} = f_{t_t}$. In these examples the significance level is set to 0.1 and is marked by the horizontal line in each plot. Bottom panel: Anomalous threshold.

$H_0 : f_{t_0} = f_{t_t}$ explained in step 6 of Algorithm 3 (top panel) and the anomalous threshold (bottom panel). In all these cases, Algorithm 3 is able to detect the occurrence of the non-stationarity right from the beginning at time point $t = 300$, while maintaining a very low FP rate (i.e., claiming the occurrence of non-stationarity when there is no actual change in the distribution) once the model has adjusted to the new distribution. As explained in Section 4.2, the anomalous threshold requires updating only if the null hypothesis $H_0 : f_{t_0} = f_{t_t}$ is rejected; that is, if a significant change in the typical behavior is detected. Thus, our proposed 'informed' approach for the detection of non-stationarity allows quicker decisions than the 'blind' approach, as it removes the requirement that the decision model be updated at each time interval.

In all of these examples, the length of the sliding window is set to 100. In each example, we obtain the initial value for the anomalous threshold by considering the first window generated by $W[1,100]$ as a representative sample of the typical behavior of the corresponding dataset.

5 Application

We apply our proposed Algorithms 1, 2 and 3 to datasets obtained using fiber optic sensor cables attached to a system. (Since the data contain commercially sensitive information, this paper does not reveal the actual application). Figure 12(a)–(c) shows the multiple parallel time series plots of three datasets. Our goal is to detect these anomalous events (such gas/oil pipeline leakages, intrusion attacks to secured premises, water contaminated areas, etc.) as soon as they start.

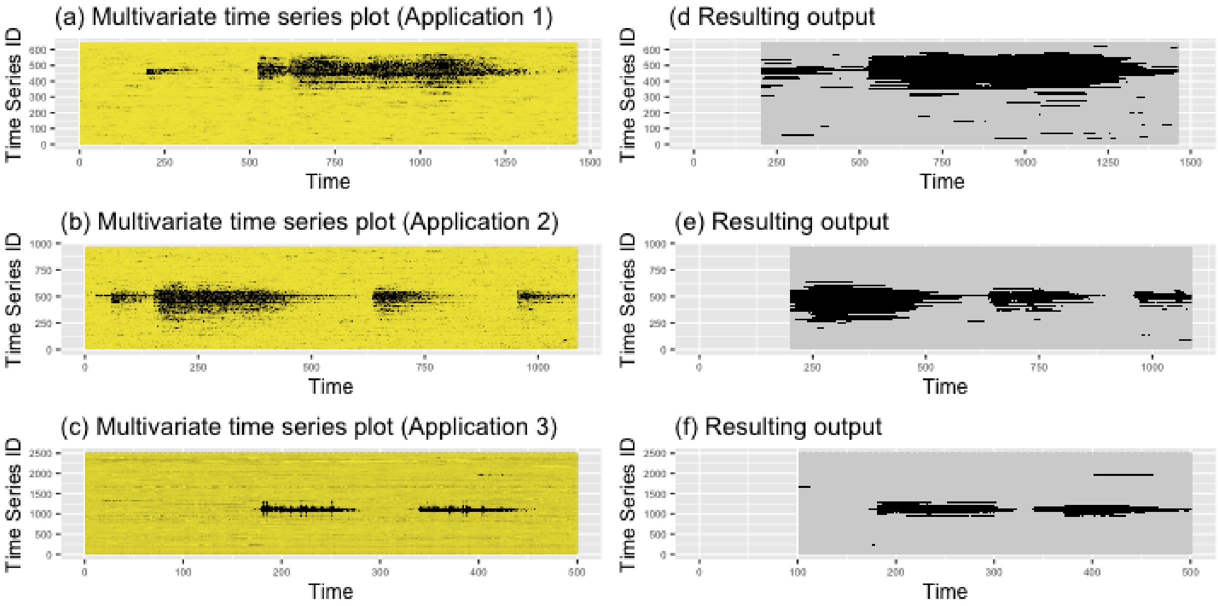

Figure 12: Application (Application 1: m = 640, Application 2: m = 1000, Application 3: m = 2500). Left panel: (black: high values, yellow: low values, black shapes are corresponding to anomalous events). Right panel: (black: outliers, gray: typical behavior)

As explained in Section 3, our proposed algorithm requires a representative sample of the typical behavior of each of these datasets in order to obtain a starting value for the anomalous threshold. However, no representative samples of the corresponding systems' typical behaviors are available for these examples. Thus, we select $W[1,100]$ for the first two examples (Figure 12(a),(b)) and $W[1,50]$ for the third example (Figure 12(c)) as the representative sample of the typical behavior in order to get an initial value for the anomalous threshold.

Even though no proper representative sample of the typical behavior was available for any of these cases, our proposed Algorithm 3 for the detection of non-stationary data distributions allows the model to adjust to the system's typical behavior over time. Figure 13 gives the cor-

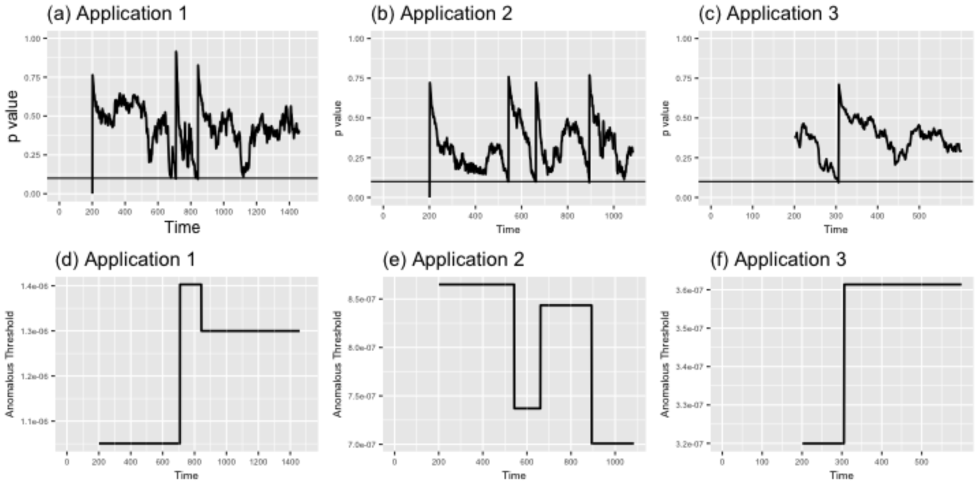

Figure 13: Detection of non-stationarity. Top panel: P value for the hypothesis test $f_{to} = f_{tt}$. In these examples the significance level is set to 0.1 and is marked by the horizontal line in each plot. Bottom panel: Anomalous threshold.

responding p-values for the hypothesis test $H_0 : f_{t_0} = f_{t_t}$ explained in step 6 of Algorithm 3 (top panel) and the anomalous threshold (bottom panel). The right panel of Figure 12 gives the output from applying Algorithms 1, 2 and 3. Since there is no "truth" for comparison, graphical representations are used to evaluate the performances of the proposed algorithms on these datasets. It can be seen from Figure 12(d)–(f) that all of the anomalous events have been captured by the proposed algorithm right from the start. The resulting outputs also follow the shapes of the actual anomalous events.

As explained in Section 4.1, here also we observe a horizontal elongation of anomalous events of the resulted outputs (Figure 12(d)–(f)) as the algorithm produces an alarm until it reaches a window that is completely free of the anomalous events. Due to this lag effect the anomalous events in the resulted outputs (Figure 12(d)–(f)) also look wider in comparison to the corresponding actual anomalous events (Figure 12(a)–(c)). However this broadening happens only in the direction of time and not in the direction of the sensor ID. This lag effect in the direction of time could be a merit for certain applications such as detection of intruders into secured premises, as the system continues to generate an alarm for certain period even after the actual event that allows to drag the attention of responsible people for necessary actions.

Although the anomalous events are correctly detected by our proposed framework, in comparison to Applications 2 and 3 (Figure 12(c), (d)), Application 1 (Figure 12(a)) shows some false positives (the isolated extra black stripes). This can be explained by Theorem 1 and Figure 3. As can be seen in Figure 12(a), Application 1 contains a small number of time series ($m \approx 600$ time series) in comparison to Applications 2 and 3. According to step 5 of Algorithm 3, in the presence of non-stationarity, the detected anomalous points are removed and only the typical points are used to update the anomalous threshold. If the detected proportion of anomalous series is high with respect to the total number of series in the collection of time series, then the new anomalous threshold could be based on a significantly different EVD (Figure 3) and thereby could lead to a higher number of false positives. But as m (the number of series in the collection) increases (as in Applications 2 and 3) the proportion of anomalous series in each window becomes very small and therefore the change in the EVD is negligible which reduces the rate of false positives as in Application 2 and 3 (Figure 12(e), (f)). Therefore, our proposed framework is particularly well suited for the applications described in Section 1, which generate large collections of time series.

6 Conclusions and Further Work

This paper proposes a methodology for the detection of anomalous series within a large collection of streaming time series using EVT. We define an anomaly here as an observation that is very unlikely given the distribution of the typical behavior of a given system. We cope with non-stationary data distributions using sliding window comparisons of feature densities, thereby allowing the decision model to adjust to the changing environment automatically as changes are detected. Our preliminary analysis using both synthetic data and data obtained using fiber optic cables reveals that the proposed framework (Algorithms 1, 2 and 3) can work well in the presence of non-stationary environments and noisy time series from multi-modal typical classes.

The density estimation in the proposed framework was done using a bivariate kernel density estimation method. Alternative methods of density estimation may lead to improved tail estimation, leading to better values for the anomalous threshold. The test of nonstationarity also depends on the kernel density estimates, and we may not reject stationarity when m is small. Log-spline bivariate density estimation (Kooperberg & Stone 1991) and local likelihood density estimation (Loader et al. 1996) would be worth considering in attempting to improve tail estimation, and thereby improve the performance of the algorithm in the presence of moderate to low values of m. In the current work, Kolmogorov-Smirnov test for the Gumbel is used to confirm the goodness-of-fit (Marshall & Olkin 2007). Alternative methods as proposed in (Clifton et al. 2014) may guide to better values for the anomalous threshold in the presence of other sub-classes of EVT.

The current framework is developed under the assumption that the measurements produced by sensors are one dimensional. The rapid advances in hardware technology has made it possible for many sensors to capture multiple measurements simultaneously, leading ultimately to a collection of multidimensional multivariate streaming time series data. An important open research problem is to extend our framework to handle such data. One possibility is to consider the features extracted from multiple measurements as a point pattern (Luca, Karsmakers & Vanrumste 2014, Luca et al. 2016, 2018) and then focus on the problem of identifying the anomalous point patterns generated by multiple measurements from individual sensors. Another possibility is to adopt a functional approach where time series of multiple measurements from individual sensors are represented by functions and anomalous thresholds are defined over the function space as in (Clifton et al. 2013).

In the current framework, the length of the sliding window is introduced as a user defined parameter that can be selected according to the application. Since the proposed framework is based on the features extracted from individual time series of a given window, a window size set too small will not be able to correctly capture the dynamic properties of the time series and thereby could reduce the performance of the framework. If, on the other hand, the window is too large, then it will take a long time to adjust to the new typical behavior in the presence of nonstationarity. Accordingly, selecting the appropriate input window size is a trade-off between classification performance and the time taken to adjust to the new typical behavior. A possible extension of the proposed framework could involve ways of optimally selecting the window size to balance this trade-off.

7 Supplemental Materials

Data and scripts: Datasets and R code to reproduce all figures in this article (main.R).

R package oddstream: The oddstream package (Talagala et al. 2018) consists of the implementation of Algorithm 1, 2 and 3 as described in this article. Version 0.5.0 of the package was used for the results presented in the article and is available from Github `https://github.com/pridiltal/oddstream`.

R-packages: Each of the R packages used in this article (*ggplot2* (Wickham 2009), *dplyr* (Wickham et al. 2017), *tibble* (Müller & Wickham 2017), *tidyr* (Wickham & Henry 2017), *reshape* (Wickham 2007)) are available online (URLs are provided in the bibliography).

Acknowledgements

This research was supported in part by the Monash eResearch Centre and eSolutions-Research Support Services through the use of the MonARCH (Monash Advanced Research Computing Hybrid) HPC Cluster. Funding was provided by the Australian Research Council through the Linkage Project LP160101885.

References

Anderson, N. H., Hall, P. & Titterington, D. M. (1994), 'Two-sample test statistics for measuring discrepancies between two multivariate probability density functions using kernel-based density estimates', *Journal of Multivariate Analysis* **50**(1), 41–54.

Bilen, C. & Huzurbazar, S. (2002), 'Wavelet-based detection of outliers in time series', *Journal of computational and graphical statistics* **11**(2), 311–327.

Burridge, P. & Taylor, A. M. R. (2006), 'Additive outlier detection via extreme-value theory', *Journal of Time Series Analysis* **27**(5), 685–701.

Catalano, A., Bruno, F. A., Pisco, M., Cutolo, A. & Cusano, A. (2014), 'An intrusion detection system for the protection of railway assets using fiber bragg grating sensors', *Sensors* **14**(10), 18268–18285.

Chandola, V., Banerjee, A. & Kumar, V. (2009), 'Anomaly detection: A survey', *ACM computing surveys (CSUR)* **41**(3), 15.

Clifton, D. A., Clifton, L., Hugueny, S. & Tarassenko, L. (2014), ‘Extending the generalised pareto distribution for novelty detection in high-dimensional spaces’, *Journal of signal processing systems* **74**(3), 323–339.

Clifton, D. A., Clifton, L., Hugueny, S., Wong, D. & Tarassenko, L. (2013), ‘An extreme function theory for novelty detection’, *IEEE Journal of Selected Topics in Signal Processing* **7**(1), 28–37.

Clifton, D. A., Hugueny, S. & Tarassenko, L. (2011), ‘Novelty detection with multivariate extreme value statistics’, *Journal of signal processing systems* **65**(3), 371–389.

Cuppens, K., Karsmakers, P., Van de Vel, A., Bonroy, B., Milosevic, M., Luca, S., Croonenborghs, T., Ceulemans, B., Lagae, L., Van Huffel, S. et al. (2014), ‘Accelerometry-based home monitoring for detection of nocturnal hypermotor seizures based on novelty detection’, *IEEE journal of biomedical and health informatics* **18**(3), 1026–1033.

Dries, A. & Rückert, U. (2009), ‘Adaptive concept drift detection’, *Statistical Analysis and Data Mining* **2**(5-6), 311–327.

Duong, T., Goud, B. & Schauer, K. (2012), ‘Closed-form density-based framework for automatic detection of cellular morphology changes’, *Proceedings of the National Academy of Sciences* **109**(22), 8382–8387.

Embrechts, P., Klüppelberg, C. & Mikosch, T. (2013), *Modelling Extremal Events: for Insurance and Finance*, Stochastic Modelling and Applied Probability, Springer Berlin Heidelberg.
URL: *https://books.google.com.au/books?id=BXOI2pICfJUC*

Faria, E. R., Gonçalves, I. J., de Carvalho, A. C. & Gama, J. (2016), ‘Novelty detection in data streams’, *Artificial Intelligence Review* **45**(2), 235–269.

Farrar, C. R. & Worden, K. (2012), *Structural health monitoring: a machine learning perspective*, John Wiley & Sons.

Fisher, R. A. & Tippett, L. H. C. (1928), Limiting forms of the frequency distribution of the largest or smallest member of a sample, *in* ‘Mathematical Proceedings of the Cambridge Philosophical Society’, Vol. 24, Cambridge Univ Press, pp. 180–190.

Fulcher, B. D. (2012), Highly comparative time-series analysis, PhD thesis, University of Oxford.

Galambos, J., Lechner, J. & Simiu, E. (2013), *Extreme Value Theory and Applications: Proceedings of the Conference on Extreme Value Theory and Applications, Volume 1 Gaithersburg Maryland 1993*, Extreme Value Theory and Applications: Proceedings of the Conference on Extreme Value Theory and Applications, Gaithersburg, Maryland, 1993, Springer US.
URL: *https://books.google.com.au/books?id=XMPkBwAAQBAJ*

Galeano, P., Peña, D. & Tsay, R. S. (2006), 'Outlier detection in multivariate time series by projection pursuit', *Journal of the American Statistical Association* **101**(474), 654–669.

Gama, J., Sebastião, R. & Rodrigues, P. P. (2013), 'On evaluating stream learning algorithms', *Machine learning* **90**(3), 317–346.

Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M. & Bouchachia, A. (2014), 'A survey on concept drift adaptation', *ACM Computing Surveys (CSUR)* **46**(4), 44.

Gupta, M., Gao, J., Aggarwal, C. & Han, J. (2014), 'Outlier detection for temporal data: A survey', *IEEE Transactions on Knowledge and Data Engineering* **26**(9), 2250–2267.

Hayes, M. & Capretz, M. (2015), 'Contextual anomaly detection framework for big sensor data', *Journal of Big Data* **2**(1), 2.

Hossin, M. & Sulaiman, M. (2015), 'A review on evaluation metrics for data classification evaluations', *International Journal of Data Mining & Knowledge Management Process* **5**(2), 1.

Hugueny, S. (2013), Novelty detection with extreme value theory in vital-sign monitoring, PhD thesis, University of Oxford.

Hyndman, R. J., Wang, E. & Laptev, N. (2015), Large-scale unusual time series detection, *in* 'Data Mining Workshop (ICDMW), 2015 IEEE International Conference on', IEEE, pp. 1616–1619.

Jiang, Q. & Sui, Q. (2009), Technological study on distributed fiber sensor monitoring of high voltage power cable in seafloor, *in* 'Automation and Logistics, 2009. ICAL'09. IEEE International Conference on', IEEE, pp. 1154–1157.

Jin, R. & Agrawal, G. (2007), Frequent pattern mining in data streams, *in* 'Data Streams', Springer, pp. 61–84.

Kang, Y., Hyndman, R. J., Li, F. et al. (2018), Efficient generation of time series with diverse and controllable characteristics, Technical report, Monash University, Department of Econometrics and Business Statistics.

Kang, Y., Hyndman, R. J. & Smith-Miles, K. (2017), 'Visualising forecasting algorithm performance using time series instance spaces', *International Journal of Forecasting* **33**(2), 345–358.

Kooperberg, C. & Stone, C. J. (1991), 'A study of logspline density estimation', *Computational Statistics & Data Analysis* **12**(3), 327–347.

Krohn, D. A., MacDougall, T. & Mendez, A. (2000), *Fiber optic sensors: fundamentals and applications*, Isa.

Lavin, A. & Ahmad, S. (2015), Evaluating real-time anomaly detection algorithms–the numenta anomaly benchmark, *in* 'Machine Learning and Applications (ICMLA), 2015 IEEE 14th International Conference on', IEEE, pp. 38–44.

Loader, C. R. et al. (1996), 'Local likelihood density estimation', *The Annals of Statistics* **24**(4), 1602–1618.

Luca, S., Clifton, D. A. & Vanrumste, B. (2016), 'One-class classification of point patterns of extremes', *The Journal of Machine Learning Research* **17**(1), 6581–6601.

Luca, S. E., Pimentel, M. A., Watkinson, P. J. & Clifton, D. A. (2018), 'Point process models for novelty detection on spatial point patterns and their extremes', *Computational Statistics & Data Analysis* **125**, 86–103.

Luca, S., Karsmakers, P., Cuppens, K., Croonenborghs, T., Van de Vel, A., Ceulemans, B., Lagae, L., Van Huffel, S. & Vanrumste, B. (2014), 'Detecting rare events using extreme value statistics applied to epileptic convulsions in children', *Artificial intelligence in medicine* **60**(2), 89–96.

Luca, S., Karsmakers, P. & Vanrumste, B. (2014), Anomaly detection using the poisson process limit for extremes, *in* 'Data Mining (ICDM), 2014 IEEE International Conference on', IEEE, pp. 370–379.

Luts, J., Broderick, T. & Wand, M. P. (2014), 'Real-time semiparametric regression', *Journal of Computational and Graphical Statistics* **23**(3), 589–615.

Ma, J. & Perkins, S. (2003), Time-series novelty detection using one-class support vector machines, *in* 'Neural Networks, 2003. Proceedings of the International Joint Conference on', Vol. 3, IEEE, pp. 1741–1745.

Mahadevan, S. & Shah, S. L. (2009), 'Fault detection and diagnosis in process data using one-class support vector machines', *Journal of process control* **19**(10), 1627–1639.

Marshall, A. W. & Olkin, I. (2007), *Life distributions*, Vol. 13, Springer.

Moshtaghi, M., Bezdek, J. C., Havens, T. C., Leckie, C., Karunasekera, S., Rajasegarar, S. & Palaniswami, M. (2014), 'Streaming analysis in wireless sensor networks', *Wireless Communications and Mobile Computing* **14**(9), 905–921.

Müller, K. & Wickham, H. (2017), *tibble: Simple Data Frames.* R package version 1.4.1.
URL: *https://CRAN.R-project.org/package=tibble*

Nikles, M. (2009), Long-distance fiber optic sensing solutions for pipeline leakage, intrusion, and ground movement detection, *in* 'Fiber optic sensors and applications VI', Vol. 7316, International Society for Optics and Photonics, p. 731602.

O'Reilly, C., Gluhak, A., Imran, M. A. & Rajasegarar, S. (2014), 'Anomaly detection in wireless sensor networks in a non-stationary environment', *IEEE Communications Surveys & Tutorials* **16**(3), 1413–1432.

Peña, D. & Prieto, F. J. (2001), 'Multivariate outlier detection and robust covariance matrix estimation', *Technometrics* **43**(3), 286–310.

Perron, P. & Rodríguez, G. (2003), 'Searching for additive outliers in nonstationary time series', *Journal of Time Series Analysis* **24**(2), 193–220.

Rajasegarar, S., Leckie, C., Bezdek, J. C. & Palaniswami, M. (2010), 'Centered hyperspherical and hyperellipsoidal one-class support vector machines for anomaly detection in sensor networks', *IEEE Transactions on Information Forensics and Security* **5**(3), 518–533.

Ranawana, R. & Palade, V. (2006), Optimized precision-a new measure for classifier performance evaluation, *in* 'Evolutionary Computation, 2006. CEC 2006. IEEE Congress on', IEEE, pp. 2254–2261.

Rapach, D. E. & Strauss, J. K. (2008), 'Structural breaks and garch models of exchange rate volatility', *Journal of Applied Econometrics* **23**(1), 65–90.

Raskutti, B. & Kowalczyk, A. (2004), 'Extreme re-balancing for svms: a case study', *ACM Sigkdd Explorations Newsletter* **6**(1), 60–69.

Riani, M., Atkinson, A. C. & Cerioli, A. (2009), 'Finding an unknown number of multivariate outliers', *Journal of the Royal Statistical Society: series B (statistical methodology)* **71**(2), 447–466.

Rodríguez, J. J. & Kuncheva, L. I. (2008), Combining online classification approaches for changing environments, *in* 'Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)', Springer, pp. 520–529.

Schwarz, K. T. (2008), *Wind dispersion of carbon dioxide leaking from underground sequestration, and outlier detection in eddy covariance data using extreme value theory*, ProQuest.

Sundaram, S., Strachan, I. G. D., Clifton, D. A., Tarassenko, L. & King, S. (2009), Aircraft engine health monitoring using density modelling and extreme value statistics, *in* 'Proceedings of the 6th International Conference on Condition Monitoring and Machine Failure Prevention Technologies'.

Talagala, P. D., Hyndman, R. J. & Smith-Miles, K. (2018), *oddstream: Outlier Detection in Data Streams*. R package version 0.5.0.
URL: *https://github.com/pridiltal/oddstream*

Wickham, H. (2007), 'Reshaping data with the reshape package', *Journal of Statistical Software* **21**(12).
URL: *http://www.jstatsoft.org/v21/i12/paper*

Wickham, H. (2009), *ggplot2: Elegant Graphics for Data Analysis*, Springer-Verlag New York.
URL: *http://ggplot2.org*

Wickham, H., Francois, R., Henry, L. & Müller, K. (2017), *dplyr: A Grammar of Data Manipulation*. R package version 0.7.4.
URL: *https://CRAN.R-project.org/package=dplyr*

Wickham, H. & Henry, L. (2017), *tidyr: Easily Tidy Data with 'spread()' and 'gather()' Functions*. R package version 0.7.2.
URL: *https://CRAN.R-project.org/package=tidyr*

Wilkinson, L. (2018), 'Visualizing big data outliers through distributed aggregation', *IEEE transactions on visualization and computer graphics* **24**(1), 256–266.

Yoon, S., Ye, W., Heidemann, J., Littlefield, B. & Shahabi, C. (2011), 'Swats: Wireless sensor networks for steamflood and waterflood pipeline monitoring', *IEEE network* **25**(1).

Zhang, Y., Meratnia, N. & Havinga, P. J. (2010), 'Ensuring high sensor data quality through use of online outlier detection techniques', *International Journal of Sensor Networks* **7**(3), 141–151.

Zhuang, L. & Dai, H. (2006), Parameter estimation of one-class svm on imbalance text classification, *in* 'Conference of the Canadian Society for Computational Studies of Intelligence', Springer, pp. 538–549.