Anomaly Detection in Streaming Nonstationary Temporal Data

Priyanga Dilini Talagala, Rob J Hyndman, Kate Smith-Miles, Sevvandi Kandanaarachchi and Mario A Muñoz

March 2018

Working Paper 4/18
Anomaly Detection in Streaming Nonstationary Temporal Data

Priyanga Dilini Talagala
Department of Econometrics & Business Statistics, Monash University, Clayton VIC 3800, Australia
Email: Dilini.Talagala@monash.edu
Corresponding author

Rob J Hyndman
Department of Econometrics & Business Statistics
Monash University, Clayton VIC 3800, Australia

Kate Smith-Miles
School of Mathematics & Statistics, University of Melbourne, Parkville VIC 3052, Australia

Sevvandi Kandanaarachchi
Department of Econometrics & Business Statistics
Monash University, Clayton VIC 3800, Australia

Mario A Muñoz
School of Mathematics & Statistics, University of Melbourne, Parkville VIC 3052, Australia

8 March 2018

JEL classification: C38, C55, C60
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 forecasts 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 time series. 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-stationarity (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.

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 a presence of an anomalous event (e.g., intrusion attack, gas pipeline leak, etc.) that has been taken place during the 500–1300 time period.

Fiber optic sensing technology can be used to detect unusual, critical events such as power cable faults, electrical short circuits, gas or oil pipeline leakages, intruders to secured premises, etc. (Krohn, MacDougall & Mendez 2000). 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 is pulsed through the cable and 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 fiber may indicate intrusions. Similarly, sensor cables can monitor temperature profiles along gas and oil pipelines, allowing the detection of leakages (Krohn, MacDougall & Mendez 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; 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 extreme value theory 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-stationarity. 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.

The ideas presented in this article have been implemented in the open source R package, oddstream. Supplementary materials available online include datasets and code used to produce the figures in this article.

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 neighbouring 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 extreme value theory.

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). The algorithm proposed by Schwarz (2008) using extreme value theory is capable of detecting both outliers of this nature and additive 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, Wang & Laptev (2015) who have proposed a method using principal component analysis applied to time series features, together with highest density regions and $\alpha$-hulls, to identify unusual time series in a large collection of time series. The recent work of Wilkinson (2018) also has the ability to address problems of this nature.

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

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, Broderick & Wand 2014). With batch processing, 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 methods proposed by Hyndman, Wang & Laptev (2015) and Wilkinson (2018) can be used to identify anomalous series within a large collection of series in the batch scenario.

The streaming data scenario poses many additional challenges, due 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 nonstationarity (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

The algorithm proposed in this paper is based on extreme value theory, a branch of probability theory that relates to the statistical behavior of extreme order statistics in a given sample (Galambos, Lechner & Simiu 2013). The Fisher-Tippett Theorem (Fisher & Tippett 1928) is the basis of classical extreme value theory, where $x_i \in \mathbb{R}$. The following expression of the theorem has been adapted from Theorem 3.2.3 of Embrechts, Klüppelberg & Mikosch (2013), p.121; the notation has been changed for consistency.

**Theorem 2.1** (Fisher-Tippett theorem, limit laws for maxima).

Let $X = \{x_1, x_2, \ldots, x_m\}$ be a sequence of independent and identically distributed random variables with cumulative distribution function (CDF) $F$ and density function $f = F'$. Let $X_{\text{max}} = \max(X)$. If there exists a centering constant $d_m (\in \mathbb{R})$ and a normalizing constant $c_m (> 0)$, and some non-degenerate distribution function $H^+ ('+' \text{ refers to the distribution of maxima})$ such that $c_m^{-1}(X_{\text{max}} - d_m) \overset{d}{\rightarrow} H^+$, then $H^+$ belongs to one of the three distribution function types: Fréchet $\Phi^+_\alpha(x)$, Weibull $\Psi^+_\alpha(x)$ or Gumbel $\Lambda^+(x)$.

Even though these distributions differ in their modelling purposes, they are closely related from a mathematical point of view. The following properties can be verified immediately (Embrechts, Klüppelberg & Mikosch 2013; Hugueny 2013): $X^{-1} \in \text{MDA}(\Psi^-_{\alpha}) \iff -X^{-1} \in \text{MDA}(\Psi^+_{\alpha}) \iff \log(X)^{\alpha} \in \text{MDA}(\Lambda^+)$ with shape parameter $\alpha$. This relationship will be referred to hereafter as **Result 1** (MDA refers to the Maximum Domain of Attraction). Interested readers are referred to the work of Embrechts, Klüppelberg & Mikosch (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 (or clusters of observations) with large nearest neighbor distances are defined as anomalies. Within this framework, the ‘spacing theorem’ (Schwarz 2008) in extreme value theory 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 (or cluster of observations) 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.
Anomaly Detection in Streaming Nonstationary Temporal Data

(2009), Clifton, Hugueny & Tarassenko (2011) and Hugueny (2013) provide a few examples where extreme value theory 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 2.1 that the extreme value distributions are parameterized implicitly by \( m \), the size of the sample from which the extrema is taken. Thus, different values of \( m \) can yield different extreme value distributions as shown in Figure 3. Clifton, Hugueny & Tarassenko (2011) proposed a numerical method for selecting a threshold for identifying anomalous points when \( m \geq 1 \). In their “\( \Psi \) transform method”, Clifton, Hugueny & Tarassenko (2011) define the “most extreme” of a set of \( m \) samples \( X = \{x_1, x_2, \ldots, x_m\} \), distributed according to pdf \( f(x) \), as the most improbable with respect to the distribution; i.e., \( \arg \min_{x \in X} \{f(x)\} \).

![Figure 3: Extreme value distributions corresponding to \( m = 1, 10, 100, 1000 \), each describing where the maximum of \( m \) samples drawn from \( N(0,1) \) will lie. Each density plot is based on 1,000 data points. (Reproduced from Clifton, 2009, p.51.)](image)

\[ \text{Figure 3: Extreme value distributions corresponding to } m = 1, 10, 100, 1000, \text{ each describing where the maximum of } m \text{ samples drawn from } N(0,1) \text{ will lie. Each density plot is based on 1,000 data points. (Reproduced from Clifton, 2009, p.51.)} \]

3 Methodology

This section proposes a new framework for anomaly detection in multivariate streaming time series based on the \( \Psi \)-transformation method proposed by Clifton, Hugueny & Tarassenko (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

First, the method proposed by Hyndman, Wang & Laptev (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.

Second, the “HDoutliers” method proposed by Wilkinson (2018) relies on the assumption that the nearest-neighbor distances of anomalous points (or clusters of 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 pressurised 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, the HDoutliers algorithm would fail to detect the anomalous results, as the series are not widely separated from the typical points in the feature space.

Our proposed method requires a representative dataset of the system’s typical behavior. The remarkable progress that has been made in data collection and storage technology has now made it possible for many organizations to accumulate large amounts of data very cheaply. 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:** $D_{\text{norm}}$, a collection of $n$ time series (which can be of either equal or different lengths) that are generated from $n$ sensors under the typical system behavior.

**Output:** $t^*$, anomalous threshold.

1. Extract $m$ features (similar to Fulcher (2012) and Hyndman, Wang & Laptev (2015)) from each of the time series in $D_{\text{norm}}$. This produces an $n \times m$ 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 allows us to handle time series of different lengths and/or starting points, as it can transform time series of any length or starting point into a vector of features of a fixed size. It 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 used in this work, eight were selected from Hyndman, Wang & Laptev (2015) (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). 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 (rmeaniqumean), 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 $n \times m$ feature matrix, $M$. Let $M^*$ represent the resulting $n \times m$ feature matrix.

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

4. Define a two-dimensional space using the first two principal components (PC) from step 3 (similar to Hyndman, Wang & Laptev (2015) and Kang, Hyndman & Smith-Miles (2017)). Each data point on this two-dimensional space corresponds to a time series in $D_{norm}$. This was done with the aim of maximizing our chances of obtaining insights via visualization (Kang, Hyndman & Smith-Miles 2017). The resulting two-dimensional space is referred to as the 2D PC space.

5. Estimate the probability density of this 2D PC space using kernel density estimation with a bivariate Gaussian kernel. Let $\hat{f}_2$ denote the estimated probability density function.

6. Draw a large number $N$ of extremes (as defined in Clifton, Hugueny & Tarassenko (2011)) from $\hat{f}_2$, and form an empirical distribution of their densities in the $\Psi$-transform space, where the $\Psi$-transform is defined as

$$\Psi[\hat{f}_2(x)] = \begin{cases} (-2\ln(\hat{f}_2(x)) - 2\ln(2\pi))^{1/2}, & \hat{f}_2(x) < (2\pi)^{-1} \\ 0, & \hat{f}_2(x) \geq (2\pi)^{-1} \end{cases}.$$ 

7. Fit a Gumbel distribution to the resulting $\Psi[\hat{f}_2(x)]$ values. The Gumbel parameter values are obtained via maximum likelihood estimation. Since the probability density values were calculated using kernel density estimation with a bi-variate Gaussian kernel, it can be shown that $\hat{f}$ falls in the maximum domain of attraction of a Weibull distribution, and can be written as $\hat{f}(x) \in \text{MDA}(\Psi_\alpha)$, where $\alpha$ is the shape parameter. (The proof of this is similar to the proof of Proposition 5 of Hugueny (2013), p.147). According to Result 1 in Section 2.3, $-\alpha\log(\hat{f}(x)) \in \text{MDA}(\Lambda^+)$. Following Clifton, Hugueny & Tarassenko (2011) and Hugueny (2013), the value of $\alpha$ is set to one, as the dimension of our feature space is equal to two. This yields $-\log(\hat{f}(x)) \in \text{MDA}(\Lambda^+)$. According to Hugueny (2013), since the $\Psi$-transform consists of taking the log of the probability density, it is in the domain of attraction of $\Lambda^+$. 

Talagala, Hyndman, Smith-Miles, Kandanaarachchi, Muñoz: 8 March 2018
Anomaly Detection in Streaming Nonstationary Temporal Data

8. Using the distribution fitted in step 7, determine the location of an anomalous threshold on $f_2^*$ by equating the corresponding CDF, $F_2^*$ (which is univariate in $\Psi$-space), to some probability mass, such as $F_2^*[f_2(x)] = 0.999$, thereby defining a contour $t^*$ in the 2D PC space that describes where the most extreme of the $n$ typical samples generated from $f_2$ will lie, to some level of probability (e.g., 0.999).

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.


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

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

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

![Figure 5: Feature based representation of the time series in Figure 1. Axis 'Cable' represents individual points of the sensor cable. There are 640 time series. Each plot is corresponding to a feature type extracted from the 640 time series. Almost all the features have captured the unusual event near the right end point of the cable (around 350 to 550).](image)
2. Project this new feature matrix, $M_{test}$, on the same 2D PC space of the typical data that was built using the time series in $D_{norm}$. Let $y_1, y_2, \ldots, y_n$ represent data points that are obtained by projecting $M_{test}$ on this 2D PC space.

3. Calculate the probability density values of $y_1, y_2, \ldots, y_n$ with respect to $f_2$ in step 5 of Algorithm 1.

4. Find any $y_j$ that satisfies $f_2(y_j) < t^*$, where $j = 1, 2, \ldots, n$, 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-stationarity

Non-stationarity can occur in many different forms. In the econometrics literature, non-stationarity is sometimes classified as either a “structural break” or a “time-varying” evolutionary change (Rapach & Strauss 2008). In the machine learning literature, non-stationarity 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, Sebastião & Rodrigues (2013), there are two approaches that can be used to adapt models in order to handle concept drift: blind and informed. Under the blind approach, the decision model is updated at regular time intervals without considering the reality (whether a change has really occurred or not). In contrast, the informed approach updates the decision model only if an occurrence of concept drift is detected (Faria et al. 2016).

Following the definition of Dries & Rückert (2009), this paper proposes an informed approach for early detection of non-stationarity that uses statistical distance measures to measure the distance between the distribution 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.

#### Algorithm 3 (Detection of non-stationarity)

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

**Output:** A vector of indexes 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 $n$ time series of length $w$. Extract $m$ features (the same features as were defined in step 1 of Algorithm 1) from each of these $n$ time series in $W[t - w, t]$. This produces an $n \times m$ feature matrix, $M_{test}$.

3. Project $M_{test}$ onto the 2D PC space of $D_0$. Let $Y_t$ represent the newly projected data points.

4. Identify the 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 $D_0$. Let $Y_{t,norm}(\subseteq Y_t)$ 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 $f_0$, the probability density function of $Y_{t,norm}$, using kernel density estimation with a bivariate Gaussian kernel. Let $\hat{f}_0$ denote the estimated probability density function.

   b) Estimate $f_t$, the probability density function of $Y_t$, using kernel density estimation with a bivariate Gaussian kernel. Let $\hat{f}_t$ denote the estimated probability density function. In the case of a ‘sudden’ change, all (or most) of the points in $Y_t$ may lie outside the anomalous boundary, defined by $D_0$. As a result, all (or most) of those points in $Y_t$ 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 $H_0 : f_0 = f_t$. Since the distributions of these distance measures are unknown, bootstrap methods can be used to determine critical points for the test (Anderson, Hall & Titterington 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, Goud & Schauer (2012), we test the null hypothesis $H_0 : f_0 = f_t$ here by using the squared discrepancy measure $T = \int [f_0(x) - f_t(x)]^2 dx$, which was proposed by Anderson, Hall & Titterington (1994). Since the test statistic based on the integrated
square distance between two kernel-based density estimates 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 $p < p^*$, $D_{t_0}^k$ is set to $W[t-w,t]_{\text{norm}}$. If $H_0$ is rejected and $p > p^*$, $D_{t_0}^k$ 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 Experiment

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.

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. The dataset is generated such that the collection of series consists of a bi-modal typical class throughout the entire period. 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. 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 further both the nature of the noisy signals and the progress and structure of the anomalous event in the 400–1000 time period. 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 $W[151,300]$. However, even though the
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.

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 false positive 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 0.989 accuracy level on average for this dataset, while maintaining low false positive (0.0075 on average) and false negative (0.003 on average) rates.
4.2 Anomaly detection with non-stationarity

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 behaviour 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 widow 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 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
Figure 8: ‘Gradual’ non-stationarity starting from $t = 300$.

Figure 9: ‘Reoccuring’ type non-stationarity starting from $t = 300$.

Figure 10: ‘Incremental’ non-stationarity starting from $t = 300$. 
Anomaly Detection in Streaming Nonstationary Temporal Data

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 false positive and false negative 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 0.95 for the entire period,
while maintaining low false positive (0.047 on average) and false negative (0.000 on average) rates during
the time period under consideration.

Figure 11: Detection of concept drift. P value for the hypothesis test $H_0 : f_{t_0} = f_t$. In these examples the
significance level is set to 0.05 and is marked by the horizontal line in each plot.

Figure 11 illustrates the change in distribution over time via the $p$-value of the hypothesis test $H_0 : f_{t_0} = f_t$
explained in step 6 of Algorithm 3. 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 false positive
rate (i.e., claiming the occurrence of non-stationarity when there is no actual change in the distribution)
one the model has adjusted to the new distribution. As explained in Section 4.2, the anomalous threshold
requires updating only if the null hypothesis $H_0 : f_{t_0} = f_t$ 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 as gas/oil pipeline leakages, intrusion attacks to secured premises, water contaminated areas, etc.) as soon as they start.

![Figure 12: Application. Left panel: (black: high values, yellow: low values, black shapes are corresponding to anomalous events). Right panel: (black: outliers, gray: typical behavior)](image)

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-stationarity allows the model to adjust to the system’s typical behavior over time. Figure 13 gives the corresponding $p$-values for the hypothesis test $H_0 : f_{t_0} = f_t$ explained in step 6 of Algorithm 3. In each of these examples, the significance level is set to 0.05 and is marked by a black horizontal line. The right panel of Figure 12 gives the output from applying Algorithms
Anomaly Detection in Streaming Nonstationary Temporal Data

Figure 13: Detection of concept drift. P value for the hypothesis test \( f_{t_0} = f_{t_1} \). In these examples the significance level is set to 0.05 and is marked by the horizontal line in each plot.

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.

6 Conclusion

This paper proposes a methodology for the detection of anomalous series within a large collection of streaming time series using extreme value theory. 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-stationarity 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-stationarity and noisy time series from multi-modal typical classes.

7 Supplemental Materials

Data and scripts: Datasets and R code to reproduce all figures in this article (main.R).
R-package oddstream: R-package oddstream consists of the implementation of Algorithm 1, 2 and 3 as described in the article. Version 0.1.0 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).

References


