MONASH University

Anomaly Detection in Streaming Time Series Data

Priyanga Dilini Talagala

B.Sc. (Hons), University of Sri Jayewardenepura, Sri Lanka

A thesis submitted for the degree of Doctor of Philosophy at

Monash University in 2019

Department of Econometrics and Business Statistics

Contents

Copyright Notice

Abstract

Anomaly detection has wide variations in problem formulations, which demand different analytical approaches. Despite the ever-increasing attention and resources devoted to the area of anomaly detection, some challenges are not supported by the existing frameworks and algorithms. This thesis reduces this gap by introducing three new algorithms for anomaly detection with special reference to their capabilities, competitive features and target applications.

This thesis offers four fundamental contributions. First, it proposes an improved algorithm for anomaly detection in high-dimensional data. It outperforms the state-of-the-art methods in many examples in terms of both accuracy and computational efficiency, while retaining a valid probabilistic interpretation for the anomalous threshold. Further, many existing algorithms have been specifically developed for the batch scenario, where it is assumed that all available data have been collected prior to analysis. However, with the recent rapid advances in data collection technology, streaming data are now becoming increasingly important and pose various challenges due to nonstationarity, noisy signals, large volume, high velocity, incomplete events and online support. To meet these challenges, as a second contribution, the thesis proposes another algorithm that provides early detection of anomalies within a large collection of streaming time series data. This algorithm includes a novel approach that adapts to nonstationarity. Third, it proposes a new algorithm to detect anomalies, caused by technical issues, in water-quality data from in situ sensors. Fourth, with the aim of facilitating reproducible research, the first, second and third algorithms are implemented in three open source R packages: `stray`, `oddstream` and `oddwater`, respectively. Using various synthetic and real datasets, this thesis demonstrates the wide applicability and usefulness of the three algorithms.

In `stray`, an anomaly is defined as an observation that deviates markedly from the majority with a large distance gap. This improved unsupervised algorithm for high-dimensional data is based on distance measures and the extreme value theory. In `oddstream`, an anomaly is defined as an observation that is very unlikely, given the recent distribution of a given system. In this algorithm, a boundary for the system's typical behaviour is calculated using the extreme value theory. Then, a sliding window is used to test newly arrived data. The model uses time series features as inputs and a density-based comparison to locate nonstationarity. `Oddwater` involves an application where anomaly detection is performed using turbidity, conductivity and river level data collected from rivers flowing into the Great Barrier Reef lagoon, Australia.

The algorithm, `stray`, which is specially designed for high-dimensional data, addresses the limitations of the state-of-art-method, the `HDoutliers` algorithm. Using various applications, this thesis demonstrates how `stray` can be used to detect anomalies in other data types, such as temporal data and streaming data. Applications of `oddstream` with data obtained using fibre optic cables showed that the framework has the ability to provide early detection of anomalies in large streaming nonstationary data. `Oddwater` successfully identified abrupt changes caused by technical outliers in water-quality sensors, while maintaining very low false detection rates.

Publications During Enrolment

This thesis by publication is built around four articles which are at different stages of publication.

1. Chapter 2 has been submitted to *Journal of Computational and Graphical Statistics* for possible publication.

   Talagala, P. D., Hyndman, R. J. & Smith-Miles, K. (2019). Anomaly Detection for High Dimensional Data. *arXiv preprint arXiv:1908.04000.*

2. Chapter 3 is published in *Journal of Computational and Graphical Statistics.*

   Talagala, P. D., Hyndman, R.J., Smith-Miles, K., Kandanaarachchi, K., & Muñoz, M.A., (2019) Anomaly Detection in Streaming Nonstationary Temporal Data, *Journal of Computational and Graphical Statistics*, DOI: 10.1080/10618600.2019.1617160.

   I won the ACEMS Business Analytics prize 2018 for this work.

3. Chapter 4 is published in *Water Resources Research.*

   Talagala, PD, RJ Hyndman, C Leigh, K Mengersen, and K Smith-Miles (2019). A feature-based procedure for detecting technical outliers in water-quality data from in situ sensors. *Water Resources Research*, DOI: 10.1029/2019WR024906.

4. Chapter 5 is published in the *Science of the Total Environment.*

   Leigh, C., Alsibai, O., Hyndman, R. J., Kandanaarachchi, S., King, O. C., McGree, J. M., Neelamraju, C., Strauss, J., Talagala, P.D., Turner, R.D., Mengersen, K., & Peterson, E.E. (2019). A framework forautomated anomaly detection in high frequency water-quality data from in situ sensors. *Science of the Total Environment* 664, 885-898.

The contribution in Chapters 2, 3 and 4 of this thesis were presented at the following events:

- 37th International Symposium on Forecasting 2017, Cairns, Australia.
- Young Statisticians Conference 2017, Tweed Heads NSW, Australia.

  A Statistical Society of Australia travel award grant was received to attend the conference.
- Young Stats Showcase hosted by the Statistical Society of Australia, Victorian Branch, Australia, in September 2017.
- 38th International Symposium on Forecasting 2018, Boulder, Colorado, USA.

  An International Institute of Forecasters travel award grant was received to attend the conference.
- 2018 Joint Statistical Meetings (JSM2018), Vancouver, British Columbia, Canada
- useR! 2018, Brisbane, Australia.
- Joint International Society for Clinical Biostatistics and Australian Statistical Conference 2018 (ISCB ASC18), Melbourne, Australia.

  I won the EJP Pitman Young Statisticians Prize 2018, Merit Award 'for the outstanding talk presented by a *Young Statistician* at an Australian Statistical Conference'.
- 39th International Symposium on Forecasting, Thessaloniki, Greece.

  An International Institute of Forecasters travel award grant was received to attend the conference.
- useR! 2019, Toulouse, France.

  A conference diversity scholarship was received to attend the conference.

Declaration

I hereby declare that this thesis contains no material which has been accepted for the award of any other degree or diploma at any university or equivalent institution and that, to the best of my knowledge and belief, this thesis contains no material previously published or written by another person, except where due reference is made in the text of the thesis.

This thesis includes two original papers published in peer reviewed journals and two submitted publications. The core theme of the thesis is anomaly detection in streaming time series data. The ideas, development and writing up of all the papers (with the exception of Chapter 5) in the thesis were the principal responsibility of myself, the student, working within the Department of Econometrics and Business Statistics, Monash University, under the supervision of Professor Rob J. Hyndman and Professor Kate Smith-Miles.

The inclusion of co-authors reflects the fact that the work came from active collaboration between researchers and acknowledges input into team-based research.

In the case of Chapter 2-5 my contribution to the work involved the following:

Thes Chap ter	Publication Title	Status (published, in press, accepted or returned for revision)	Nature and % of student contribution	Co-author name(s) Nature and % of Co-author's contribution	Co-author(s) Monash student Y/N
2	Anomaly Detection for High Dimensional Data	Submitted	Formulating the approach, construction of research design, implementation, data analysis and interpretation, software development, writing the first draft (80%)	1) Rob J. Hyndman, input into manuscript (10%) 2) Kate Smith-Miles, input into manuscript (10%)	N
3	Anomaly Detection in Streaming Non-stationary Temporal Data	Published	Formulating the approach, construction of research design, implementation, data analysis and interpretation, software development, writing the first draft (80%)	1) Rob J. Hyndman, input into manuscript (10%) 2) Kate Smith-Miles, input into manuscript (5%) 3) Sevvandi Kandanaarachchi and Mario A. Muñoz, input into manuscript (5%)	N
4	A feature-based procedure for detecting technical outliers in water-quality data from in situ sensors	Published	Formulating the approach, construction of research design, implementation, data analysis and interpretation, software development, writing the first draft (80%)	1) Rob J. Hyndman, input into manuscript (10%) 2) Catherine Leigh, Kerrie Mengersen and Kate Smith-Miles, input into manuscript (10%)	N
5	A framework for automated anomaly detection in high frequency water-quality data from in situ sensors	Published	Formulating the feature based anomaly detection approach: implementation, data analysis and interpretation, software development, writing the first draft of the feature based approach related parts. Developing a Shiny web application to explore data. Data preprocessing. (30%)	(1) Catherine Leigh, input into manuscript (40%) 2) Rob J. Hyndman, input into manuscript (10%) 3) Other co-authors, input into manuscript (20%)	N

I have not renumbered sections of submitted or published papers in order to generate a consistent presentation within the thesis.

Student name: Priyanga Dilini Talagala

Date: 23.08.2019

Acknowledgements

Over the past few years, with my experience on anomaly detection, I realised how important the support from the surrounding points is for a point to stand out as an anomaly, just like my support system around me supported me to make this thesis a reality. Now, it is my great pleasure to thank everyone who made this thesis possible.

I am deeply grateful to my supervisor Professor Rob J. Hyndman. Your valuable guidance and your way of thinking about, and doing, research continue to inspire me and shaped my thinking about all facets of this research. My research style, my research focus, my research toolkit – they all have their own roots in your mentorship. I am also very grateful to my co-supervisor Professor Kate Smith-Miles from University of Melbourne, Australia, who always raised important questions I never would have considered. I am truly blessed to have you two throughout this very special journey. Your influence on my thinking about research from different angles and boosting my confidence is beyond estimation.

Chapters 3 and 4 are based on the collaborative research project carried out with the Queensland University of Technology and the Queensland Department of Environment and Science, Great Barrier Reef Catchment Loads Monitoring Program. I consider myself very lucky to have had such a great opportunity to work on this project with many wonderful mentors, including Professor Kerrie Mengersen, Dr Erin Peterson and Dr. Catherine Leigh. Thank you for your thoughtful critiques, support, guidance, encouragement and generous hospitality during my visit to Queensland University of Technology, Australia, in April 2018. It was pure joy working with you all.

I was able to learn so much at, Monash University because of financial support from the Monash Graduate Scholarship and the Faculty Graduate Research scholarship. This

assistance has provided me many unique opportunities that I will always be thankful for. I am also thankful to ARC Centre of Excellence for Mathematical and Statistical Frontiers (ACEMS) for funding my visit to Queensland to work on the water-quality project and the Australian Research Council through the Linkage Project LP160101885 for funding the research study in Chapter 1 of this thesis. I am also thankful to the Queensland Department of Environment and Science; in particular, the Great Barrier Reef Catchment Loads Monitoring Program for the data, and the staff from Water Quality and Investigations for their input on Chapters 3 and 4 of this thesis. Further, 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. I want to thank them for their valuable contribution.

I am also very thankful to my thesis committee, Professor Gael Martin, Professor Farshid Vahid, Professor George Athanasopoulos and Professor Xueyan Zhao, for their valuable feedback and suggestions. I am also grateful to David Hill and the many anonymous reviewers who read the manuscripts presented in Chapters 3, 4 and 5 and generously provided insights and suggestions to improve our work. I am also thankful to all my co-authors for their valuable contribution and collaborations. Special thanks to Cathy Morgan and Elite Editing for helping in copy editing and clarifying the manuscripts. The editorial intervention by Elite Editing was restricted to Standards D and E of the Australian Standards for Editing Practice.

I take this opportunity to thank the Statistical Society of Australia (SSA) for the travel grant to present at the Young Statisticians Conference 2017, Tweed Heads, NSW, Australia; the International Institute of Forecasters for the travel grant to present at the 38th International Symposium on Forecasting, Boulder, Colorado, USA and the travel grant to present at the 39th International Symposium on Forecasting, Thessaloniki, Greece; useR! 2019 for the travel grant to present at useR! 2019, Toulouse, France; and Monash EBS travel grant to present at JSM 2018, Vancouver, Canada. Each of these conferences is a unique experience and allowed me to share my research findings with a wider community. These conferences helped me significantly to attract the attention of peers and experts in

my field, and the questions, comments and suggestions I received from both academic and industry-oriented researchers helped me considerably in improving my work.

During my stay at Monash, the department administrative staff members, especially Clare Livesey, were very supportive and positive regarding every request. The Monash graduate research team was also very helpful and timely with all requests and I want to thank them for their assistance.

Special thanks to my sister, Thiyanga Talagala, who was there with, and for me, every step of the way, sharing many hats as siblings, schoolmates, college mates, workmates and office mates. I was extremely lucky to have an opportunity to have her by my side and share our latest and most precious hats as academic sisters and batch mates together at Monash University. Deeply grateful to our parents whom I love, admire, respect and find comfort in beyond words; you are a pillar of strength for me. All three of you have had a bigger influence on this thesis than you might realise.

Chapter 1

Introduction

Anomaly detection is an important research topic that has been explored within diverse research areas and application domains. The presence of anomalies in data can lead to biased parameter estimation, model misspecification and misleading results if classical analysis techniques are blindly applied (Abuzaid, Hussin, and Mohamed, 2013; Ben-Gal, 2005). Conversely, anomalies themselves can be the main carriers of significant and often critical information and the identification of these critical points can be the main purpose of many investigations in fields such as fraud detection (e.g., credit card frauds and network intrusion), object tracking (e.g., flight tracking), system health monitoring (e.g., machine breakdown and power cable leakages) and environmental monitoring (e.g., water quality, bushfire, earthquake and volcanic eruption) (Gupta et al., 2014). Further, owing to rapid advances in data collection technology it has become increasingly common for organisations to be dealing with data that stream in large quantities. Therefore, the overall focus of this thesis is on detecting anomalies in streaming time series data.

1.1 Background

Anomaly detection problems have many different facets, and the detection techniques can be highly influenced by the way we define anomalies, the type of input data to the algorithm, the expected output, etc. This leads to wide variations in problem formulations, which need to be addressed through different analytical approaches. A number of surveys

of anomaly detection techniques have been done in general (Singh and Upadhyaya, 2012; Aggarwal, 2017) or for specific data domains such as temporal data (Gupta et al., 2014), streaming data (Habeeb et al., 2019), network data (Ranshous et al., 2015; Shahid, Naqvi, and Qaisar, 2015; Kwon et al., 2017), graph data (Akoglu, Tong, and Koutra, 2015), tensor data (Fanaee-T and Gama, 2016), intrusion detection (Mitchell and Chen, 2014) and novelty detection (Pimentel et al., 2014). This section however limits the review to the background work on anomaly detection for streaming time series data and lays the foundation for the work presented in Chapters 2–5.

1.1.1 Definitions Found in the Literature

Solutions to the problem of detecting unusual behaviours in systems of interest can be influenced heavily by the way in which anomalies are defined. Three terms are used commonly and interchangeably in the literature to describe work related to the topic: *novelty* (Clifton, Hugueny, and Tarassenko, 2011; Hugueny, 2013), *anomaly* (Hyndman, Wang, and Laptev, 2015; Kumar et al., 2016) and *outlier* (Schwarz, 2008; Wilkinson, 2018). However, Faria et al. (2016) differentiate between these three terms, using the terms anomaly and outlier to refer to the idea of an undesired pattern but novelty to refer to the emergence of a new concept that needs to be incorporated into the typical behaviour of the system. In line with this view, Chandola, Banerjee, and Kumar (2009) define an anomaly as a pattern in the data that does not conform to the expected behaviour but a novelty as an unobserved pattern that is typically incorporated into the model of the typical behaviour of a given system when it is detected. However, Gama (2010) points out that a substantial number of examples is required as evidence of the appearance of a novelty before it should be incorporated into the model of the typical behaviour of a given system. Thus, the sparse examples that differ considerably from the 'typical' behaviour can all be considered anomalies or outliers, since there is no guarantee that they represent a new 'typical' behaviour of the system (Faria et al., 2016). Lavin and Ahmad (2015) define anomalies in streaming data, with respect to their past behaviour, as patterns that do not conform to the past behaviours of the system. As a result, a new behaviour may be anomalous at first, but it ceases to be anomalous if the new 'typical' pattern continues to exist, and ultimately ends up being a novelty rather than an anomaly or an outlier.

Grubbs (1969) defines an anomaly as an observation that deviates markedly from other members of the sample. However, this deviation can be defined in terms of either distance or density. Burridge and Taylor (2006), Wilkinson (2018) and Schwarz (2008) have all proposed methods for anomaly detection by defining an anomaly in terms of distance. In contrast, Hyndman (1996), Clifton, Hugueny, and Tarassenko (2011) and Hugueny (2013) have proposed methods that define an anomaly with respect to either the density or the chance of the occurrence of observations.

1.1.2 Representations of Time Series

Fulcher and Jones (2014) consider two representations of time series: instance-based and feature-based.

The instance-based representation of time series is the most straightforward and has been used by many researchers in the data mining community. Under this representation, if two time series are to be compared, a distance measure between the two time series is defined that leads to a direct comparison of the ordered values of the two time series. The methods proposed by Wilkinson (2018), Clifton, Hugueny, and Tarassenko (2011) and Hugueny (2013) are all based on this representation of time series.

In contrast to the instance-based representation of time series, the feature-based representation of time series involves representing a given time series in terms of its properties, measured using different statistical operations, thereby transforming a temporal problem into a static problem (Fulcher, Little, and Jones, 2013). After extracting features, further analysis is based on these extracted features. Thus, this representation can allow an algorithm to compare time series of different lengths and/or starting points, because it can transform time series of any length or starting point into a vector of features of a fixed size. Recently, researchers such as Wang, Smith, and Hyndman (2006), Fulcher (2012) and Hyndman, Wang, and Laptev (2015) have paid a considerable amount of attention to the feature-based representation of time series, since it helps to reduce the dimension of the original multivariate time series problem via features that encapsulate the dynamic properties of the individual time series efficiently.

1.1.3 Extreme Value Theory

The algorithms proposed in Chapters 2, 3 and 4 are based on th extreme value theory, a branch of probability theory that relates to the behaviour of extreme order statistics in a given sample (Galambos, Lechner, and Simiu, 2013). In contrast to traditional data analysis, where the primary focus is on the observations in the central region of the distribution, extreme value theory focuses primarily on modelling the distribution of extreme order statistics in a given sample (Pinto and Garvey, 2016; Clifton, 2009). The central limit theorem is one of the most striking limit theorems in statistics. Its ability to approximate the distribution of the sample mean irrespective of the parent distribution of the original random variable is the property that makes this theorem so remarkable (Coles, 2001). Analogous arguments are used in the extreme value theory to approximate distributions of extreme order statistics in a given sample.

1.1.4 Key Results of Classical Extreme Value Theory

Consider a set of m independent and identically distributed (iid) data, $\mathbf{X} = \{x_1, x_2, \ldots, x_m\}$, which has its own cumulative distribution function (CDF), F, and an associated probability density function (pdf), f. In classical extreme value theory, $x_i \in \Re$ (univariate). Let $X_{max} = max(\mathbf{X})$ and $X_{min} = min(\mathbf{X})$. The extreme value theory focuses on the statistical behaviour of these quantities. Hereafter, the discussion will focus on X_{max} (X_{min} will be referred to only when necessary), because it simplifies the discussion, but a similar argument can be applied to X_{min} as well.

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. Figure 1.1 (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, because 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), because it describes the expected location of the maximum of a sample of size m generated from f (Clifton, Hugueny, and Tarassenko, 2011). The Fisher–Tippett Theorem (Fisher and Tippett, 1928), which is the basis of classical extreme value theory, explains the possibilities for this H^+.

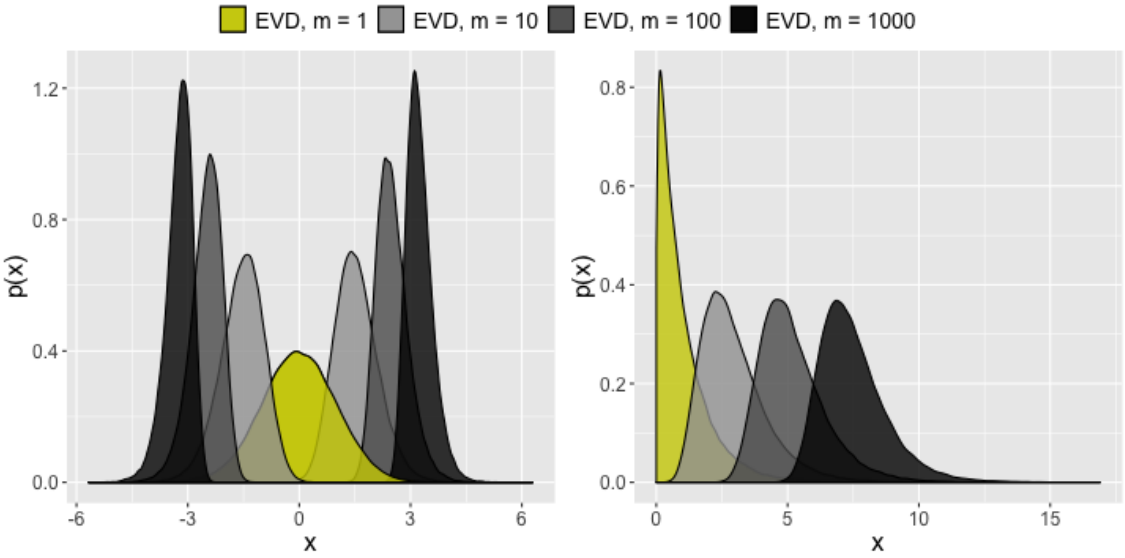


Figure 1.1: *Empirical distributions of* 10^6 *minima and maxima for the standard Gaussian distribution (left), and of maxima for the standard exponential distribution (right). (Reproduced from Hugueny, 2013, p.87.)*

Theorem 1.1 (Fisher-Tippett theorem, limit laws for maxima)**.** *(Theorem 3.2.3 in* Embrechts, Klüppelberg, and Mikosch (2013)*, p. 121; the notations have been changed for consistency within this thesis.)*

Let $\boldsymbol{X} = \{x_1, x_2, \ldots, x_m\}$ *be a sequence of iid random variables and* $X_{max} = max(\boldsymbol{X})$*. If there exists a centring constant* $d_m (\in \Re)$ *and normalising 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 following three distribution function types:*

$$\textit{Fréchet}: \quad \Phi_\alpha^+(x) = \begin{cases} 0, & x \leqslant 0 \\ exp\{-x^{-\alpha}\}, & x > 0 \end{cases} \qquad \alpha > 0$$

$$\textit{Weibull}: \quad \Psi_\alpha^+(x) = \begin{cases} exp\{-(-x)^{\alpha}\}, & x \leqslant 0 \\ 1, & x > 0 \end{cases} \qquad \alpha > 0$$

$$\textit{Gumbel}: \quad \Lambda^+(x) = exp\{-e^{-x}\}, \qquad x \in \Re.$$

Definition 1.1 (Extreme value distribution and extremal random variable)**.** (Definition 3.2.6 in Embrechts, Klüppelberg, and Mikosch (2013), p. 124)

The distribution functions $\Phi_\alpha, \Psi\alpha$ and Λ as presented in Theorem 1.1 are called standard extreme value distributions and the corresponding random variables, standard extremal random variables. Distribution functions of the types of $\Phi_\alpha, \Psi\alpha$ and Λ are extreme value distributions; the corresponding random variables are extremal random variables.

□

From Theorem 1.1, it can be observed that the extreme value distributions are implicitly parameterised by m, the size of the sample from which the extrema is taken. Therefore, different values of m will yield different extreme value distributions (Clifton, Hugueny, and Tarassenko, 2011).

Definition 1.2 (Maximum domain of attraction)**.** (Definition 3.3.1 in Embrechts, Klüppelberg, and Mikosch (2013), p. 128; the notations have been changed for consistency within this thesis.)

We say that the rv X (the distribution function F of X or the distribution of X) belongs to the maximum domain of attraction of the extreme value distribution H^+ if there exist constants $c_n > 0$, $d_n \in \Re$ such that:

$$c_m^{-1}(X_{max} - d_m) \xrightarrow{d} H^+.$$

We write $X \in MDA(H^+)$ $(F \in MDA(H^+))$.

□

The following properties, highlighted by Embrechts, Klüppelberg, and Mikosch (2013), will assist in deciding the maximum domain of attraction of the three extreme value distributions to which X belongs. Let $x_F = \sup\{x \in \Re : F(x) < 1\}$ denote the right endpoint of F.

- All distribution functions $F \in MDA(\Phi_\alpha^+)$ have an infinite right endpoint $x_F = \infty$ (the tail decreases like a power low). The Pareto, F, Cauchy and log-gamma distribution functions are just a few examples covered by the maximum domain of attraction of the Fréchet distribution.

- All distribution functions $F \in MDA(\Psi_\alpha^+)$ have a finite right endpoint $x_F < \infty$ (truncated tail). The uniform and beta distributions are two examples covered by the maximum domain of attraction of the Weibull distribution.

- Unlike the Fréchet and Weibull distributions, the maximum domain of attraction of the Gumbel distribution is not easy to characterise, because all distribution functions $F \in MDA(\Lambda^+)$ can have either a finite or an infinite endpoint $x_F \leqslant \infty$. Perhaps one way of thinking of the maximum domain of attraction of the Gumbel distribution is that it consists of distribution functions whose right tail decreases to zero faster than any power function (exponentially decaying tail). The exponential, gamma, normal and lognormal distributions are just a few examples covered by the maximum domain of attraction of the Gumbel distribution.

Extreme value distributions for minima can be discussed in a similar manner. In Chapter 3, we are particularly interested in the Weibull extreme value distribution for minima, which is given by

$$\Psi_\alpha^-(x) = \begin{cases} 0, & x < 0 \\ 1 - exp\{-x^{-\alpha}\}, & x \geqslant 0 \end{cases}$$

where '-' refers to the distribution of minima.

Interested readers are referred to the work of Embrechts, Klüppelberg, and Mikosch (2013) for a detailed discussion of the characterisation of the three classes: Gumbel, Fréchet and Weibull.

Definition 1.3 (Quantile function)**.** (Definition 3.3.5 in Embrechts, Klüppelberg, and Mikosch (2013), p.130)

The generalised inverse of the distribution function F

$$F^{\leftarrow}(t) = \inf\{x \in \Re : F(x) \geqslant t\}, \quad 0 < t < 1,$$

is called the quantile function of the distribution function F. The quantity $x_t = F^{\leftarrow}(t)$ defines the t-quantile of F.

□

Theorem 1.2 (Maximum domain of attraction of Ψ_α^-)**.** *(Theorem 1 in Clifton, Hugueny, and Tarassenko (2011), p. 384; the notations have been changed for consistency within this thesis)*

The distribution function F belongs to the maximum domain of attraction of the minimal Weibull distribution (Ψ_α^-), $\alpha > 0$, if and only if $x_F > -\infty$ and $F(x_F + x^{-1}) = x^{-\alpha}L(x)$ for some slowly varying function L.

If $F \in MDA(\Psi_\alpha^-)$, then

$$c_m^{-1}(X_{min} - x_F) \underset{\rightarrow}{d} \Psi_\alpha^-,$$

where the normalising constant c_m and the centring constant d_m can be chosen as $c_m = x_F + F^{\leftarrow}(m^{-1})$ and $d_m = x_F$. X_{min} is the minimum of m data. $x_F = inf\{x \in \Re : F(x) \leqslant 0\}$. $F^{\leftarrow}(t)$ is the t-quantile of F. L is a slowly varying function at ∞; that is, a positive function for all $t > 0$ that obeys

$$lim_{x \longrightarrow \infty} \frac{L(tx)}{L(x)} = 1.$$

Although these extreme value distributions differ in their purposes for modelling, they are related closely from a mathematical point of view. The following properties can be verified immediately (Embrechts, Klüppelberg, and Mikosch, 2013; Hugueny, 2013):

$$\begin{aligned} & X^{-1} \in MDA(\Psi_\alpha^-) \text{ with shape parameter } \alpha \\ \iff \quad & -X^{-1} \in MDA(\Psi_\alpha^+) \text{ with shape parameter } \alpha \\ \iff \quad & ln(X)^\alpha \in MDA(\Lambda^+). \end{aligned}$$

□

Let $X_1, X_2, \ldots, X_n$ be a sample from a distribution function F and let $X_{1:n} \geqslant X_{2:n} \geqslant \cdots \geqslant X_{n:n}$ be the order statistics. The available data are $X_{1:n}, \ldots, X_{k:n}$ for some fixed k.

Theorem 1.3 (Spacing theorem)**.** *(Proposition 1 in Burridge and Taylor (2006), p. 6 and Theorem 3 in Weissman (1978), p. 813; the notations have been changed for consistency in this thesis.)*

Let $D_{i,n} = X_{i:n} - X_{i+1:n}$, $(i = 1, \ldots, k)$ be the spacing between successive order statistics. If F is in the maximum domain of attraction of the Gumbel distribution, then spacings $D_{i,n}$ are asymptotically independent and exponentially distributed with mean proportional to i^{-1}.

This theorem is illustrated using Figure 1.2, which shows the distribution of the descending order statistics $(X_{i:n})$ and the standardized spacings, $(iD_{i,n})$, for $i \in \{1, \ldots, 10\}$ for $1,000$ samples each containing $20,000$ random numbers from the standard normal distribution. Figure 1.2 (a) shows the distribution of $X_{i:n}$ with means of $X_{i:n}$ depicted as black crosses. The gaps between consecutive black crosses give the spacings between higher-order statistics $(D_{i,n})$. We note that the normal distribution is in the maximum domain of attraction of the Gumbel distribution and that this example contains no outliers. A consequence of Theorem 1.3 is that the standardised spacings $(iD_{i,n})$ for $(i = 1, \ldots, K)$, are approximately iid (Burridge and Taylor, 2006). Figure 1.2 (b) shows the distribution of the standardised spacings $(iD_{i,n})$ for $(i = 1, 2, \ldots, 10)$ for $1,000$ samples of size $20,000$. Each letter-value

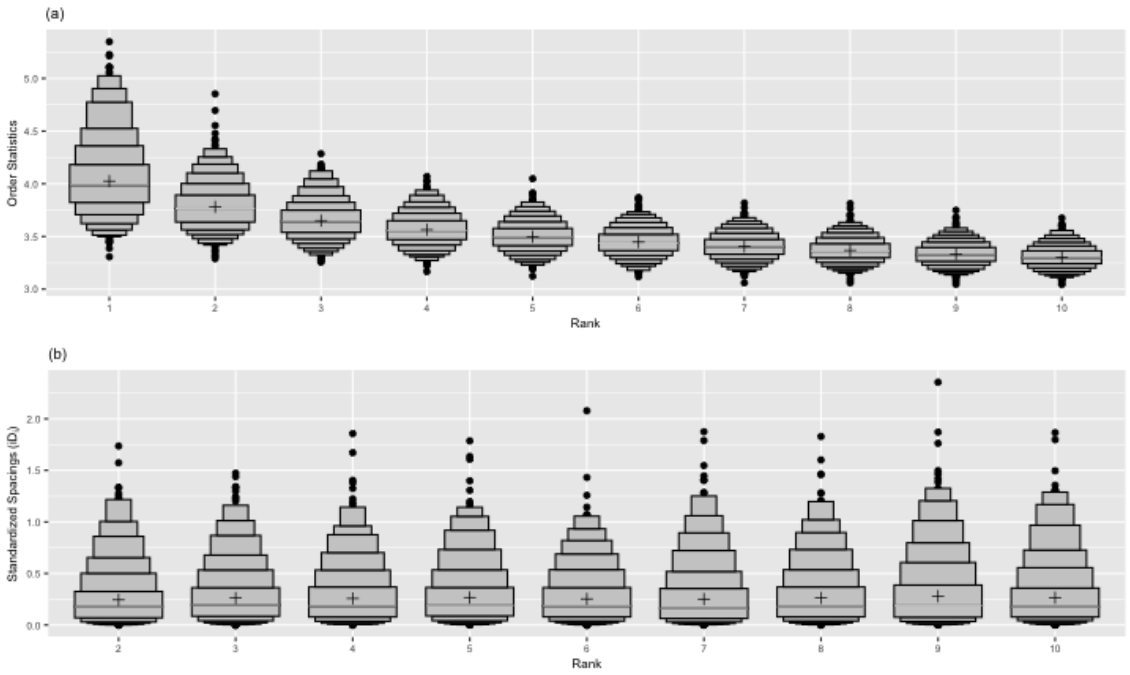


Figure 1.2: *(a) Distribution of the descending order statistics $X_{i:n}$ and (b) distribution of the standardised spacings $iD_{i,n}$ for $i \in \{1, \ldots, 10\}$ for $1,000$ samples each containing $20,000$ random numbers from the standard normal distribution.*

box plot (Hofmann, Wickham, and Kafadar, 2017) exhibits approximately the shape of an exponential distribution.

1.1.5 Calculation of Anomalous Threshold

Chapter 2 and Chapter 4 both use Theorem 1.3 (Spacing Theorem by Weissman (1978)) to estimate a data-driven anomalous threshold to discriminate anomalies. This step sorts anomalous scores and searches for any large gap at the upper tail of the distribution defined by the anomalous scores. This search for significant gaps in the upper tail can either be performed using the top-down algorithm by Burridge and Taylor (2006) or bottom-up algorithm by Schwarz (2008).

Top-down algorithm

As the name implies, the top-down algorithm introduced by Burridge and Taylor (2006) starts from the maximum and moves backwards over the sorted array, seeking a significantly large gap (which may not exist in a typical data set free from outliers). As summarised by Schwarz (2008), the top-down algorithm is as follows:

- Let $X_1, X_2, \ldots X_n$ be a sample from a distribution function F, and let $X_{1:n} \geqslant X_{2:n} \geqslant \cdots \geqslant X_{n:n}$ be the order statistics.
- Let $D_i = X_{i:n} - X_{i+1:n}$ be the spacing between successive order statistics.
- Calculate the standardised spacings, $S_i \equiv iD_i$.
- Find the maximum of the first N/α spacings, S_k, where N is the maximum possible number of outliers and α is the acceptable false positive rate. The quantity N/α then represents the number of spacings that must be examined to achieve a significance level of α.
- If $k \leq N$ spacings, mark the top k values as anomalies.
- In addition to the gap between anomalous points and the valid data, sometimes there can be multiple gaps in between different groups of anomalous points. Therefore, repeat the above steps on the remaining data until no more gaps are found in the top N values.

However, repeating the process over data until it detects all the discrete groups of anomalies present in the dataset makes the algorithm inefficient for massive datasets with vast amounts of data. Further, it is not desirable to set a value for the maximum possible number of outliers, because this value is not known in advance for many real-world applications. Ideally, the algorithm should be able to pick all the anomalies present in the data without having this predetermined number. Further, according to Schwarz (2008), this algorithm does not use the full power of the spacing theorem; it only employs the fact that the standardised spacings are iid and fails to use the fact that they are exponentially distributed, which could have given more information about how unlikely is a given spacing. The bottom-up algorithm introduced by Schwarz (2008) has the ability to release these unrealistic assumptions and overcome the limitations of the top-down algorithm. The bottom-up algorithm is based on the work of Burridge and Taylor (2006) but uses the full power of the spacing theorem.

Bottom-up algorithm

As in the top-down algorithm, the bottom-up algorithm is also based on the assumption that anomalies can bring large separations between valid data and anomalies, compared

with the separations between valid data among themselves. However, in contrast to the top-down algorithm, the bottom-up algorithm now starts from the middle of the sorted data array, which represents the valid data, and moves forward towards the upper tail of the sorted array until it reaches a large gap, which is highly unlikely to occur if it is generated from the same distribution of the valid data. When a gap is encountered that is well beyond expectation, it terminates the searching process and marks all the points above that value as outliers. The specific steps of the bottom-up algorithm proposed by Schwarz (2008) is as follows:

- Let $X_1, X_2, \ldots X_n$ be a sample from a distribution function F, and let $X_{1:n} \geqslant X_{2:n} \geqslant \cdots \geqslant X_{n:n}$ be the order statistics.
- Calculate $D_i = X_{i:n} - X_{i+1:n}$, the spacing between successive order statistics.
- At each rank i, test the hypothesis that $X_{i:n}$ is the largest valid data point in the sample, with the help of the spacings immediately below it $(D_{i+1}, D_{i+2}, ..)$. If $X_{i:n}$ is the largest valid data point in the sample, according to the spacing theorem, spacings $D_i, D_{i+1}, D_{i+2}, \ldots$ should be proportional to $1, \frac{1}{2}, \frac{1}{3}, \ldots$, and so on. This allows us to use spacings $D_{i+1,n}, D_{i+2,n}, \ldots, D_{i+k,n}$ to predict the spacing D_i:

  $$\hat{D}_i = \tfrac{1}{k-1} \textstyle\sum_{j=2}^{k} j D_{i+j-1}$$

  (Since the spacing theorem applies only to a small fraction of the data ranked near the upper tail, the entire dataset cannot be used to estimate D_i and therefore is used only $k(\ll n)$ number of spacings for the estimation process. The value k should be large enough to obtain a stable estimate for D_i, but small compared with the sample size, n. Schwarz (2008) has recommended 50 spacings as a rough guideline for the value k and the same has been used by Wilkinson (2018) for large samples). If they all represent valid points, then all the terms in the summation have the same mean, which is similar to the mean of D_i. Therefore, $\hat{D}_i$ serves as an estimator for D_i.
- As in the spacing theorem, since D_i follows an exponential distribution with mean proportional to i^{-1}, for a given significance level α, a threshold t that will not be exceeded by valid data can be obtained using:

  $$t = \hat{D}_i log(1/\alpha)$$

- Work upward towards the upper tail of the sorted data array. At the first i where spacing D_i exceeds threshold t, terminate the searching process and flag $X_{i:n}$ and all the points above in the sorted array as outliers.

The bottom-up algorithm has been used in the research presented in Chapter 2 and Chapter 4 because of its obvious advantages over the top-down algorithm.

1.2 Motivation and Objectives

In light of the increasing demand for accurate and powerful automated methods for early detection of anomalies in the streaming data scenario and the lack of attention paid to this topic, the primary motivation of this thesis is to develop methods for early detection of anomalies in the streaming data context.

The **first** motivation of this thesis arises from the recently proposed HDoutliers method by Wilkinson (2018). The HDoutliers algorithm is a powerful algorithm with a strong theoretical foundation for anomaly detection in high-dimensional data. However, some limitations significantly hinder its performance level. The effect of these limitations is a tendency to increase the rate of false positives and/or rate of false negatives under certain conditions. Therefore, the first objective is to propose solutions to these limitations of the HDoutliers algorithm. Chapter 2 addresses this objective. The proposed algorithm, the stray algorithm, is based on distance measures and the extreme value theory. Chapter 2 also demonstrates how the stray algorithm can assist in detecting anomalies in other data structures, such as time series data and streaming temporal data. The improved algorithm is implemented in the open source R package `stray`.

The **second** motivation of this thesis originated from the limited research attempts on detecting anomalous series within a large collection of series in the streaming data scenario where data flow rapidly in a continuous manner. A few researchers (Hyndman, Wang, and Laptev, 2015; Wilkinson, 2018) have developed methods to identify anomalous series within a large collection of series, mainly focusing on the batch scenario where it is assumed that the entire dataset is available prior to analysis. However, in contrast to the batch scenario, the streaming data scenario poses many different challenges owing to its complex

nature evolving over time. In addition to the obvious difficulties caused by the large volume and velocity of streaming data, highly noisy signals can increase the related complexity. Nonstationarity (concept drift) is another major topic in the streaming data analysis that makes it difficult to distinguish new typical behaviour from anomalous events (Faria et al., 2016). To address this issue, detectors should be able to learn and adapt according to the conditions present. Early detection of anomalies as soon as they start but before they end is another major requirement of most applications related to this problem. Therefore, the second objective of this study is to develop a powerful automated method to detect anomalous series within a large collection of series in the streaming data context such that it meets these requirements. Chapter 3 is dedicated to achieving this objective. This chapter presents a new algorithm based on density modelling and the extreme value theory. To cope with nonstationarity (concept drift), a density-based comparison approach is proposed. The proposed algorithm can detect significant changes in the typical behaviour and automatically update the anomalous threshold upon detecting a nonstationarity. The proposed algorithm is implemented in the open source R package `oddstream`.

The **third** motivation of this thesis arises owing to the non-existence of a customised method to detect technical anomalies in high-frequency water-quality data from *in situ* sensors. Automated *in situ* sensors have the potential to revolutionise the way we manage and monitor environmental settings, such as air, soil and water. The data produced by these sensors enable us to identify fine-scale patterns, trends and extremes over space and time. Although they represent cutting-edge technology, the data they produce are still prone to errors because of many reasons, such as miscalibration, biofouling and battery failures (Horsburgh et al., 2015). Moreover, these anomalies and the ability to detect them can differ according to the geographic characteristics of the environmental system and the spatial placement of the sensors. To ensure data quality, we need to automate the real-time detection of anomalies. Therefore, our third objective is to propose a new framework for automated anomaly detection in high-frequency water-quality data from *in situ* sensors. Chapters 4 and 5 address this objective. In these chapters, an attempt was made to develop methods that can incorporate the correlation structure of several measurements taken from each site. This involves an application performing anomaly detection using turbidity,

conductivity and river level data collected from rivers flowing into the Great Barrier Reef lagoon, Australia. The proposed algorithm is implemented in the open source R package `oddwater`.

Conclusions are drawn in Chapter 6 with a discussion on potential extensions to the proposed algorithms introduced in Chapters 2, 3 and 4.

These three main objectives guided the structuring and development of the major chapters of this thesis. Since this is a thesis by publication that has an introductory chapter and a concluding chapter with articles in between, the reader may notice some amount of repetition among chapters. Each article should be self-contained and therefore has been published with relevant materials for completeness.

Chapter 2

Anomaly Detection for High Dimensional Data

This article has been submitted to *Journal of Computational and Graphical Statistics* for possible publication.

Anomaly Detection in High Dimensional Data

Priyanga Dilini Talagala[1,3]
and
Rob J. Hyndman[1,3]
and
Kate Smith-Miles[2,3]

[1]Department of Econometrics and Business Statistics, Monash University, Australia
[2] School of Mathematics and Statistics, University of Melbourne, Australia
[3] ARC Centre of Excellence for Mathematics and Statistical Frontiers (ACEMS), Australia

November 27, 2019

Abstract

The HDoutliers algorithm is a powerful unsupervised algorithm for detecting anomalies in high-dimensional data, with a strong theoretical foundation. However, it suffers from some limitations that significantly hinder its performance level, under certain circumstances. In this article, we propose an algorithm that addresses these limitations. We define an anomaly as an observation that deviates markedly from the majority with a large distance gap. An approach based on extreme value theory is used for the anomalous threshold calculation. Using various synthetic and real datasets, we demonstrate the wide applicability and usefulness of our algorithm, which we call the stray algorithm. We also demonstrate how this algorithm can assist in detecting anomalies present in other data structures using feature engineering. We show the situations where the stray algorithm outperforms the HDoutliers algorithm both in accuracy and computational time. This framework is implemented in the open source R package `stray`.

Keywords: Extreme value theory, High dimensional data, Nearest neighbour searching, Temporal data, Unsupervised outlier detection

1 Introduction

The problem of anomaly detection has many different facets, and detection techniques can be highly influenced by the way we define anomalies, type of input data and expected output. These differences lead to wide variations in problem formulations, which need to be addressed through different analytical techniques. Although several useful computational methods currently exist, developing new methods for anomaly detection continues to be an active, attractive interdisciplinary research area owing to different analytical challenges in various application fields, such as environmental monitoring (Talagala, Hyndman, Leigh, Mengersen & Smith-Miles 2019, Leigh et al. 2019,), object tracking (Gupta et al. 2014, Sundaram et al. 2009), epidemiological outbreaks (Gupta et al. 2014), network security (Hyndman et al. 2015, Cao et al. 2015) and fraud detection (Talagala, Hyndman, Smith-Miles, Kandanaarachchi & Muñoz 2019). Ever-increasing computing resources and advanced data collection technologies that emphasise real-time, large-scale data are other reasons for this growth since they introduce new analytical challenges with their increasing size, speed and complexity that demand effective, efficient analytical and computing techniques.

Anomaly detection has two main objectives, which are conflicting in nature: One downgrades the value of anomalies and attempts eliminating them, while the other demands special attention be paid to anomalies and root-cause analysis be conducted. The presence of anomalies in data can be considered data flaws or measurement errors that can lead to biased parameter estimation, model misspecification and misleading results if classical analysis techniques are blindly applied (Ben-Gal 2005, Abuzaid et al. 2013). In such situations, the focus is to find opportunities to remove anomalous points and thereby improve both the quality of the data and results from the subsequent data analysis (Novotny & Hauser 2006). In contrast, in many other applications, anomalies themselves are the main carriers of significant and often critical information, such as extreme environmental conditions (e.g., bushfire, tsunami, flood, earthquake, volcanic eruption and water contamination), faults and malfunctions (e.g., flight tracking and power cable tracking) and fraud activities (Ben-Gal 2005), that can cause significant harm to valuable lives and assets if not detected and treated quickly.

High-dimensional datasets exist across numerous fields of study (Liu et al. 2016). Some anomaly detection algorithms also use feature engineering as a dimension reduction technique and thereby convert other data structures, such as a collection of time series using time series features (Talagala, Hyndman, Leigh, Mengersen & Smith-Miles 2019, Hyndman et al. 2015), collection of scatterplots using scagnostics (Wilkinson et al. 2005) and genomic micro arrays and chemical compositions in biology (Liu et al. 2016) into high-dimensional data prior to the detection process for easy control. Under the high-dimensional data scenario, all attributes can be of the same data type or a mixture of different data types, such as categorical or numerical, which has a direct impact on the implementation and scope of the algorithm. Much research attention has been paid to anomaly detection for numerical data (Breunig et al. 2000, Tang et al. 2002, Jin et al. 2006, Gao et al. 2011). Limited methods are available that treat both numerical and categorical data using correspondence analysis, for example, as in Wilkinson (2017).

High-dimensional anomalies can arise in all the attributes or a subset of the attributes (Unwin 2019). If all anomalies in a high-dimensional data space were anomalies in a lower dimension, then anomaly detection can be performed using axis parallel views or by incorporating an additional step of variable selection for the detection process. However, in practice, certain high-dimensional instances are only perceptible as anomalies if treated as high-dimensional problems and the correlation structure of all the attributes considered. Otherwise, these tend to be overlooked if attributes are considered separately (Wilkinson 2017, Ben-Gal 2005).

The problem of anomaly detection has been extensively studied over the past decades in many application domains. Several surveys of anomaly detection techniques have been conducted in general (Chandola et al. 2009, Aggarwal 2017) or for specific data domains such as high-dimensional data, network data (Shahid et al. 2015), temporal data (Gupta et al. 2014), machine learning and statistical domains (Hodge & Austin 2004), novelty detection (Pimentel et al. 2014), intrusion detection (Sabahi & Movaghar 2008) and uncertain data (Aggarwal & Yu 2008). Some algorithms are application specific and take advantage of the underlying data structure or other domain-specific knowledge (Talagala, Hyndman, Leigh, Mengersen & Smith-Miles 2019). More general algorithms without domain-specific

knowledge are also available with their own strengths and limitations (Breunig et al. 2000, Tang et al. 2002, Jin et al. 2006, Gao et al. 2011). Among the many possibilities, the HDoutliers algorithm, recently proposed by Wilkinson (2017), is a powerful unsupervised algorithm, with a strong theoretical foundation, for detecting anomalies in high-dimensional data. The study presented by Talagala, Hyndman, Leigh, Mengersen & Smith-Miles (2019) also verifies its performances through a thorough comparative evaluation of existing state-of-the-art anomaly detection methods. Although this algorithm has many advantages, a few characteristics hinder its performance. In particular, under certain circumstances it tends to increase the rate of false negatives (i.e., the detector ignores points that appear to be real anomalies) because it uses only the nearest-neighbour distances to distinguish anomalies. Further, to deal with large datasets with numerous observations it uses the Leader algorithm (Hartigan & Hartigan 1975), which forms several clusters of points in one pass through the dataset using a ball of a fixed radius. By incorporating this clustering method, it tries to gain the ability to identify anomalous clusters of points. However, in the presence of very close neighbouring anomalous clusters it tends to increase the rate of false negatives. Further, this additional step of clustering has a serious negative impact on the computational efficiency of the algorithm when dealing with large datasets.

Through this study, we make three fundamental contributions. First, we propose an algorithm called stray, representing '**S**earch and **TR**ace **A**nomal**Y**', that addresses the limitations of the HDoutliers algorithm. The stray algorithm presented here focuses specifically on fast, accurate anomalous score calculation using simple but effective techniques for improved performance. Second, we introduce an R (R Core Team 2019) package, `stray` (Talagala, Hyndman & Smith-Miles 2019), that implements the stray algorithm and related functions. Third, we demonstrate the wide applicability and usefulness of our stray algorithm, using various datasets.

Our improved algorithm, stray, has many advantages: (1) It can be applied to both one-dimensional and high-dimensional data. (2) It is unsupervised in nature and therefore does not require training datasets for the model-building process. (3) The anomalous threshold is a data-driven threshold and has a valid probabilistic interpretation because it is based on the extreme value theory. (4) By using k-nearest neighbour distances for

anomalous score calculation, it gains the ability to deal with the masking problem. (5) It can provide near real-time support to datasets that stream in large quantities owing to its use of fast nearest neighbour searching mechanisms. (6) It can deal with data that may have multimodal distributions for typical data instances. (7) It produces both score (to indicate how anomalous the instances are) and binary classification (to reduce the searching space during the visual and root-cause analysis) for each data instance as an output. (8) It can detect outliers as well as inliers.

The remainder of this paper is organised as follows. Section 2 presents the related work to lay the foundation for the stray algorithm. Section 3 describes the limitations of the HDoutliers algorithm that hinder its performance. Section 4 presents the improved algorithm, stray, that addresses the limitations of the HDoutliers algorithm. Section 5 presents a comprehensive evaluation, illustrating the key features of the stray algorithm. Section 6 includes an application of stray algorithm related to pedestrian behaviour in the city of Melbourne, Australia. Section 7 concludes the article and presents future research directions.

2 Background

2.1 Types of Anomalies in High Dimensional Data

The problems of anomaly detection in high-dimensional data are threefold (Figure 1), involving detection of: (a) global anomalies, (b) local anomalies and (c) micro clusters or clusters of anomalies (Goldstein & Uchida 2016). Most of the existing anomaly detection methods for high-dimensional data can easily recognise global anomalies since they are very different from the dense area with respect to their attributes. In contrast, a local anomaly is only an anomaly when it is distinct from, and compared with, its local neighbourhood. Madsen (2018) introduces a set of algorithms based on a density or distance definition of an anomaly, which mainly focuses on local anomalies in high-dimensional data. Micro clusters or clusters of anomalies may cause masking problems. Very little attention has been paid to this problem relative to the other two categories. The recently proposed HDoutliers algorithm (Wilkinson 2017) addresses this problem to some extent by grouping

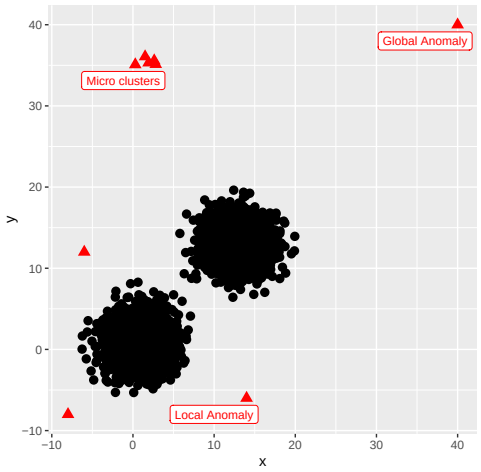


Figure 1: Different types of anomalies in high-dimensional data. Anomalies are represented by red triangles and black dots correspond to the typical behaviour.

instances together that are very close in the high-dimensional space and then selecting a representative member from each cluster before calculating nearest neighbour distances for the selected instances. In this study, we focus on all three of these anomaly types.

2.2 Definitions for Anomalies in High Dimensional Data

Anomalies are often mentioned in the literature under several alternative terms, such as outliers, novelty, faults, deviants, discordant observations, extreme values/cases, change points, rare events, intrusions, misuses, exceptions, aberrations, surprises, peculiarities, odd values and contaminants, in different application domains (Chandola et al. 2009, Gupta et al. 2014, Zhang et al. 2010). Of these, the two terms anomalies and outliers are used commonly and interchangeably in the literature describing research related to the topic. The term inlier also relates to the topic, but rarely appears in the literature on anomaly detection. Inliers are those points that appear between typical clusters without attaching to any of the clusters, but still lie within the range defined by the typical clusters (Jouan-Rimbaud et al. 1999) (Figure 2). In contrast, the corresponding notion of an 'outlier' is generally used to refer to a data instance that appears out of the space more towards the tail of a distribution, defined by the typical data instances. Some classical methods related to the topic fail to detect inliers and only focus on outliers (Jouan-Rimbaud et al. 1999).

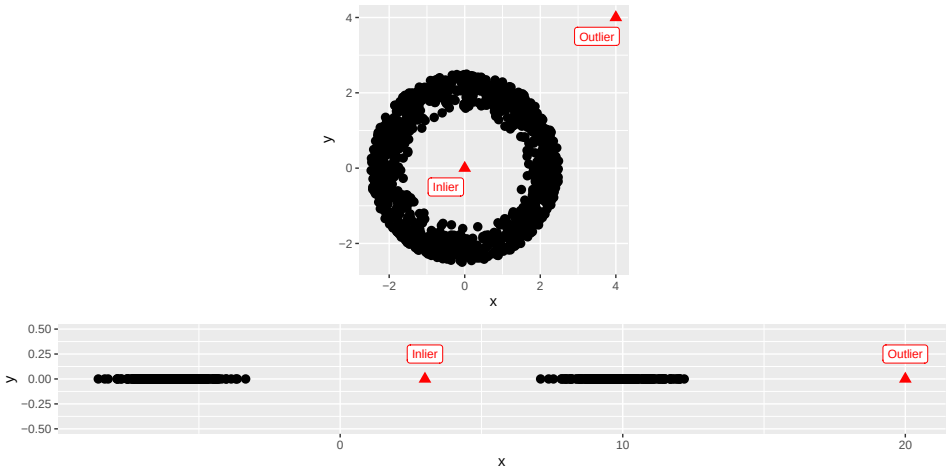


Figure 2: Inliers vs outliers. Anomalies are represented by red triangles and black dots correspond to the typical behaviour.

However, detecting inliers is equally important because they can give rise to interpolation errors. In this study, we focus on both inliers and outliers. To avoid any confusion, we use the term 'anomaly' for the purpose of nomenclature throughout this paper.

Owing to the complex nature of the problem, it is difficult to find a unified definition for an anomaly and the definition often depends on the focus of the study and the structure of the input data available to the system (Williams 2016, Unwin 2019). However, there are some definitions that are general enough to cope with datasets with various application domains. Grubbs (1969) defines an anomaly as an observation that deviates markedly from other members of the dataset. However, this deviation can be defined in terms of either distance or density. Burridge & Taylor (2006), Wilkinson (2017) and Schwarz (2008) have all proposed methods for anomaly detection by defining an anomaly in terms of distance. In contrast, Hyndman (1996), Clifton et al. (2011) and Talagala, Hyndman, Smith-Miles, Kandanaarachchi & Muñoz (2019) have proposed methods that define an anomaly with respect to either the density or the chance of the occurrence of observations. Madsen (2018) also provides a series of distance and density-based anomaly detection algorithms.

In this study, we define an anomaly as an observation that deviates markedly from the majority with a large distance gap under the assumption that there is a large distance between typical data and the anomalies compared with the distance between typical data.

2.3 HDoutliers Algorithm (Wilkinson 2017)

The HDoutliers algorithm is a distance based anomaly detection algorithm. One important property of this algorithm is that it has an ability to convert any higher dimensional anomaly detection problem to a one dimensional problem by taking the nearest neighbour distances of the data instances. There are two published versions of the HDoutliers algorithm (Wilkinson 2017), summarised below.

Algorithm 1 (HDoutliers Algorithm - Version 1)

Input: D, $n \times p$ matrix with n data instances where each data instance is p dimensional ($p \in \mathbb{Z}^+$) .

Output: A vector of indices of the anomalous data instances in D

1. Normalize the columns of D. Let D^* represent the resulting $n \times p$ matrix.
2. Compute nearest neighbour distances between all pairs of points in D^*.
3. Sort the resulting nearest neighbour distances and search for any large gap at the upper tail of the distribution and thereby define an anomalous threshold. This search for a significant gap in the upper tail is based on extreme value theory.
4. Flag data instances as anomalies using the anomalous threshold.

Algorithm 1 is recommended for small samples. The default maximum size of D for Algorithm 1 is set to 10000 in the R implementation of the `HDoutliers` package. The second version of the HDoutliers algorithm incorporates a clustering step with the aim of detecting micro clusters.

Algorithm 2 (HDoutliers Algorithm - Version 2)

Input: D, $n \times p$ matrix with n data instances where each data instance is p dimensional ($p \in \mathbb{Z}^+$) .

Output: A vector of indices of the anomalous data instances in D

1. Normalize the columns of D. Let D^* represent the resulting $n \times p$ matrix.
2. Apply the Leader algorithm (Hartigan & Hartigan 1975). Let $r = (.1/(\log n)^{1/p}$, be the radius for the Leader algorithm. Based on the value of the radius, the algorithm clusters the data instances using their nearest neighbour distances.

3. Select one data instance from each cluster as a representative member for that cluster. The algorithm selects the first element of each cluster as the representative member for that cluster. Further analysis is carried out using only those representative members, M.
4. Compute nearest neighbour distances between all pairs of points in M.
5. Sort the resulting nearest neighbour distances and search for any large gap at the upper tail of the distribution and thereby define an anomalous threshold. This search for a significant gap in the upper tail is based on extreme value theory.
6. Flag points in M as anomalies using the anomalous threshold.
7. Flag all data instances in the anomalous clusters (from step 6) as anomalous data instances.

3 Limitations of HDoutliers Algorithm

Although the HDoutliers algorithm (Wilkinson 2017) has many advantages, a few characteristics limit its possibilities. Next, we discuss these limitations in detail.

3.1 HDoutliers Uses Only the Nearest Neighbour Distance to Discriminate Anomalies

The HDoutliers algorithm uses the Leader algorithm (Hartigan & Hartigan 1975) to form small clusters of points, prior to calculating nearest neighbour distance. In the Leader algorithm, each cluster is a ball in the high-dimensional data space. In the HDoutliers algorithm, the radius of this ball is selected such that it is well below the expected value of the distances between $n(n-1)/2$ pairs of points distributed randomly in a d-dimensional unit hypercube.

After forming clusters using the Leader algorithm, the HDoutliers algorithm selects representative members from each cluster. It then calculates the nearest neighbour distances for each of these representative members. These distances are then used to identify the anomalies based on the assumption that anomalies bring large distance separations

between typical data and the anomalies, in comparison to the separations between typical data themselves. Therefore, under this assumption it is believed that any anomalous cluster will appear far away from the clusters of the typical data points. As a result, the nearest neighbour distance for this anomalous cluster will be significantly higher than that of the clusters of typical data and thereby identify it as an anomalous cluster. All the data points contained in the anomalous cluster are then marked as anomalous points within a given dataset.

However, one further assumption for this method to work properly is that any anomalous clusters present in the dataset are isolated. For example, imagine a situation in which two anomalous clusters are very close to one another but are far away from the rest of the typical clusters. Now, the two clusters will become nearest neighbours to one another and they will jointly protect them by being anomalous by giving very small nearest neighbour distances for both clusters that are compatible with the nearest neighbour distances of the rest of the typical clusters. Figures 7 (c-II) and (d-II) further elaborate this argument. In these two examples, the HDoutliers algorithm (with the clustering step) declares points as anomalies only if they are isolated and fails to detect anomalous clusters that share a few cluster neighbours. Although the HDoutliers algorithm incorporates the clustering step with the aim of identifying anomalous clusters of points, because of the very small size of the ball that is used to produce clusters (exemplars) in the d-dimensional space, it fails to bring all the points into a single cluster and instead produces a few anomalous clusters that are very close to one another. These anomalous clusters then become nearest neighbours to one another and have very small nearest neighbour distances for the representative member of each cluster. Since the detection of anomalies entirely depends on these nearest neighbour distances and since the anomalous clusters do not show any significant deviation from typical clusters with respect to the nearest neighbour distances, the algorithm now fails to detect these points as anomalies and thereby increases the rate of false negatives.

3.2 Problems Due to Clustering Via Leader Algorithm

After forming clusters of data points, the HDoutliers algorithm completely ignores the density of the data points. Once it forms clusters of data points using the Leader algorithm,

it selects a representative member from each cluster and carries out further analysis only using these representative members. Figure 7 (e-II) provides an example related to this issue. This dataset is a bimodal dataset with an anomalous point located between the two typical classes. The entire dataset contains 2,001 data points. The data points gathered at the leftmost upper corner represent one typical class with 1,000 data points. The second typical class of data points is gathered at the rightmost bottom corner with another 1,000 data points. Since this second class of data points is closely compacted in substance, the 1,000 data points are now wrapped by a single ball when forming clusters using the Leader algorithm. In the HDoutliers algorithm, the next step is to select one member from each of these clusters. Once it selects a representative member from this ball that contains 1,000 data points, it ignores the remaining 999 data points in detecting anomalies. This step misleads the algorithm, and the remaining steps of the algorithm view this representative member as an isolated data point, although it is surrounded by 999 neighbouring data points in the original dataset. Therefore, all data points in this entire class are declared as anomalies by the algorithm, although it contains half of the dataset. Unwin (2019) suggests jittering not as a perfect solution, but as an alternative to mitigate this problem. Unwin (2019) also argues that the problem tends not to occur in high-dimensional data spaces where this kind of granularity is less likely. However, then it gives rise to the problem of neighbouring anomalous clusters (as illustrated in Figure 7 (c-II, d-II)), which individually appear to be typical, or of limited suspicion (due to the presence of other neighbouring anomalous clusters), yet, their co-occurrence is highly anomalous.

Figure 7 (f-II) provides another situation in which false negatives increase because of the clustering step. This bivariate dataset contains 1,001 data points. The data points gathered at the leftmost upper corner represent a typical class covering 1,000 data points, and the isolated data point at the rightmost bottom corner represents an anomaly. Since this typical class of 1,000 data points is closely compacted, it gives rise to only 14 clusters through the Leader algorithm. Altogether, the dataset forms 15 clusters with the one created by the isolated point located at the rightmost bottom corner. Even though the original dataset contains 1,001 data points, the algorithm considers only 15 data points (a representative member from each cluster) for calculating the anomalous threshold. Now,

this number is not large enough to yield a stable estimate for the anomalous threshold. Due to this ignorance of the density of the original dataset, it now fails to detect the obvious anomalous point at the leftmost bottom corner.

3.3 Problem with Threshold Calculation

A companion R package (Fraley 2018) is available for the algorithm proposed by Wilkinson (2017). According to the R package implementation, the current version of the HDoutliers algorithm uses the next potential candidate for anomalies in calculating the anomalous threshold, in each iteration of the bottom-up searching algorithm. This approach causes an increase in the false detection rate under certain circumstances. We avoid this limitation in our proposed algorithm.

4 Proposed Improved Algorithm: stray Algorithm

In this section, we propose an improved algorithm for anomaly detection in high dimensional data. Our proposed algorithm is intended to overcome the limitations of the HDoutliers algorithm and thereby enhance its capabilities.

4.1 Input to the stray Algorithm

An input to the stray algorithm is a collection of data instances where each data instance can be a realisation of only one attribute or a set of attributes (also referred to using terms such as features, measurements and dimensions). In this study, we limit our discussion to quantitative data; therefore, an input can be a vector, matrix or data frame of $d(\geq 1)$ numerical variables, where each column corresponds to an attribute and each row corresponds to an observation of these attributes. The focus is then to detect anomalous instances (rows) in the dataset.

4.2 Normalise the Columns

Since the stray algorithm is based on the distance definition of an anomaly, nearest neighbour distances between data instances in the high-dimensional data space are the key

information for the algorithm to detect anomalies. However, variables with large variance can exert disproportional influence on Euclidean distance calculations (Wilkinson 2017). To make the variables of equivalent weight, the columns of the data are first normalised such that the data are bounded by the unit hypercube. This normalisation is commonly referred as *min-max normalisation*, which involves a linear transformation of the original data, with the result data ranging from 0 to 1 (Figures 4 (b,e). In addition to min-max normalisation, a robust normalisation method ($(x - median(x)/IQR(x))$) is also available through the `stray` package implementation. However, there is no one-fit-for-all normalisation strategy for anomaly detection problems even though min-max normalisation is shown to be preferred to median-IQR with most of the datasets and anomaly detection methods considered in Kandanaarachchi et al. (2018).

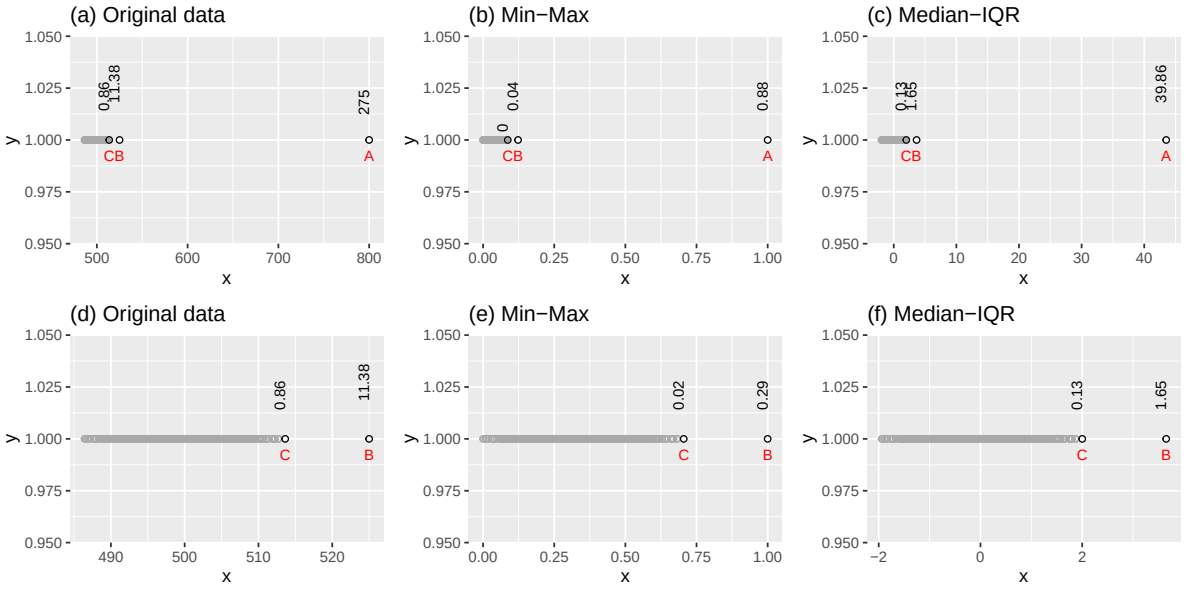


Figure 3: Effect of normalisation on nearest neighbour distance calculation. Left column: Original data set. Middle column: Min-max normalisation. Right column: Median-IQR standardization. Nearest neighbour distances are marked for the points with the highest nearest neighbour distances. Top panel: Data set with an obvious global anomaly (point A). Bottom panel: Data set without point A.

The relevance and importance of an anomalous point is determined subjectively by humans by the presence and positioning of other anomalous points. For instance, point

A in Figure 3 (a) is an obvious global anomaly and with respect to point A, point B is a trivial anomaly. In contrast, the similar point B in Figure 3 (d) appears to be nontrivial with respect to the remaining typical points. Since median-IQR standardization is robust to anomalies the nearest neighbour distances of the remaining points are not heavily influenced by the presence of obvious global anomalies (such as point A) (Figures 3 (c,f)). In contrast min-max normalisation mimics the human observational pattern by changing the nearest neighbour distances according to the presences of other nontrivial anomalies (Figures 3 (b, e)). This prevents the trivial points (such as point B in Figures 3 (b)) to emerge as local anomalies and generating false positives. However, when datasets are free from anomalies, min-max normalisation tends to increase the rate of false positives by assigning relatively high nearest neighbour distances to the boundary points even though they are a part of the typical behaviour.

4.3 Nearest Neighbour Searching

In the stray algorithm, after the columns of the dataset are normalised, it calculates the k-nearest neighbour distance with the maximum gap for each and every instance. By using this measure, we were able to address the aforementioned limitations of the HDoutliers algorithm.

For each individual observation, the algorithm first calculates the k-nearest neighbour distances, $d_{i,KNN}$, where $i = 1, 2, ..., k$. Then, it calculates the successive differences between distances, $\Delta_{i,KNN}$. Next, it selects the k-nearest neighbour distance with the maximum gap, $\Delta_{i,max}$. Figure 4 illustrates how these steps help our improved algorithm to detect anomalous points or anomalous clusters of points.

In Figure 4 (a), the dataset contains only one anomaly at $(15, 16.5)$. For this dataset, the nearest neighbour distance can differentiate the anomalous point from the remaining typical points because the nearest neighbour distance for the anomalous point is significantly larger (14.8) than that for the remaining typical points. Figure 4 (b) shows the change in the k-nearest neighbour distances of the anomaly at $(15, 16.5)$. For this dataset, the k-nearest neighbour distance with the maximum gap occurs when $k = 1$. The second dataset, in Figure 4 b), has three anomalies around $(15, 16.5)$. If we calculate only the nearest

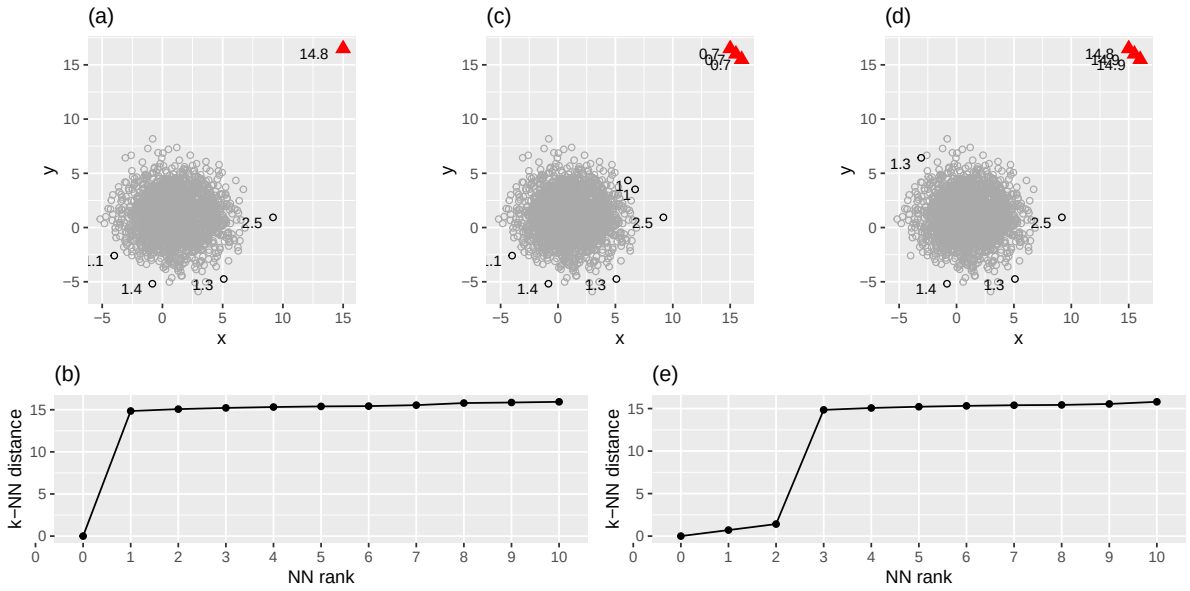


Figure 4: Difference between the nearest neighbour distance and the k-nearest neighbour distance with the maximum gap. (a) Dataset contains only one anomaly at $(15, 16.5)$. Nearest neighbour distances are marked. (b) Change in the k-nearest neighbour distances of the anomaly. (c) Dataset contains micro cluster around $(15, 16.5)$. Nearest neighbour distances are marked. (d) Dataset contains micro cluster around $(15, 16.5)$. For the three anomalies, the third nearest neighbour distance has the maximum gap. (e) Change in the k-nearest neighbour distances of an anomaly from micro cluster around $(15, 16.5)$. Anomalies are represented by red triangles and black dots correspond to the typical behaviour.

neighbour distances for each observation, then the three anomalies are not distinguishable from the typical points since their values are very small (0.7) compared with that of most typical points with nearest neighbour distances at around (0.0015 to 2.5). However, the three anomalies are distinguishable from their typical points with respect to the k-nearest neighbour distances with the maximum gap (Figure 4 (d)). For the three anomalies in Figure 4 (d), the third nearest neighbour distance has the maximum gap (Figure 4 e)) and the three points are now easily distinguished as anomalies, with respect to k-nearest neighbour distances with the maximum gap. Therefore, by using k-nearest neighbour distances with the maximum gap, the stray algorithm gains the ability to detect both anomalous singletons and micro clusters. Through this approach, we are able to reduce

the false detection rate and thereby address the limitations of the HDoutliers algorithm, while gaining the ability to detect micro clusters. This is also a very simple, but clever, investment as compared with the time taken by the leader algorithm to form small clusters to detect micro clusters (especially for datasets with large dimensions), in the HDoutliers algorithm. Further, for each point, the corresponding k-nearest neighbour distances with the maximum gap act as an anomalous score to indicate the degree of being an anomaly.

In the current study, we consider both exact and approximate k-nearest neighbour searching techniques. Brute force search involves going through every possible paring of points to detect k-nearest neighbours for each data instance, and therefore, exact k-nearest neighbours are explored. Conversely, k-dimensional trees (k-d trees) employ spatial data structures that partition space to allow efficient access to a specified query point (Elseberg et al. 2012*a*). Therefore, it involves searching approximate k-nearest neighbours around a specified query point.

In the current algorithm, parameter k, which determines the size of the neighbourhood, is introduced as a user-defined parameter that can be selected according to the application. One way to interpret the role of k in the stray algorithm is to view it as the minimum possible size for a typical cluster in a given dataset. If the size of an anomalous cluster is less than k, it will be detected as a micro cluster by the stray algorithm. The choice of k has different effects across different dimensions and sizes of data (Campos et al. 2016). We can set k to 1 if no micro clusters are present in the dataset and thereby focus on local and global anomalous points. High k values are recommended for datasets with high dimensions because of the curse of dimensionality.

4.4 Threshold Calculation

Anomalous scores assign each point a degree of being an anomaly. However, for certain applications it is also important to categorise typical and anomalous points for the subsequent root-cause analysis. Ideally, we prefer a universal threshold to unambiguously distinguish anomalous points from typical points. Following Schwarz (2008), the HDoutliers algorithm (Wilkinson 2017) defines an anomalous threshold based on extreme value theory, a branch of probability theory that relates to the behaviour of extreme order statistics in a given

sample (Galambos et al. 2013).

The anomalous threshold calculation in Schwarz (2008); Burridge & Taylor (2006) and Wilkinson (2017) is an application of Weissman's spacing theorem (Weissman 1978) (Theorem 1) that is applicable to the distribution of data covered by the maximum domain of attraction of a Gumbel distribution. This requirement is satisfied by a wide range of distributions, ranging from those with light tails to moderately heavy tails that decrease to zero faster than any power function (Embrechts et al. 2013). Examples include the exponential, gamma, normal and log-normal distributions with exponentially decaying tails.

Let $X_1, X_2, \ldots, X_n$ be a sample from a distribution function F and let $X_{1:n} \geqslant X_{2:n} \geqslant \cdots \geqslant X_{n:n}$ be the order statistics. The available data are $X_{1:n}, \ldots, X_{k:n}$ for some fixed k.

Theorem 1 (Spacing Theorem) . *(Proposition 1 in Burridge & Taylor (2006), p.6 and Theorem 3 in Weissman (1978), p.813; the notations have been changed for consistency in this paper)*
Let $D_{i,n} = X_{i:n} - X_{i+1:n}$, $(i = 1, \ldots, k)$ be the spacing between successive order statistics. If F is in the maximum domain of attraction of the Gumbel distribution, the spacings $D_{i,n}$ are asymptotically independent and exponentially distributed with mean proportional to i^{-1}.

We illustrate this theorem using Figure 5, which shows the distribution of the descending order statistics $(X_{i:n})$ and the standardised spacings, $(iD_{i,n})$, for $i \in \{1, \ldots, 10\}$ for $1,000$ samples each containing $20,000$ random numbers from the standard normal distribution. Figure 5 (a) shows the distribution of $X_{i:n}$ with means of $X_{i:n}$ depicted as black crosses. The gaps between consecutive black crosses give the spacings between higher-order statistics $(D_{i,n})$. We note that the normal distribution is in the maximum domain of attraction of the Gumbel distribution and that this example contains no outliers. A consequence of Theorem 1 is that the standardised spacings $(iD_{i,n})$ for $(i = 1, \ldots, K)$, are approximately iid (Burridge & Taylor 2006). Figure 5 (b) shows the distribution of the standardised spacings $(iD_{i,n})$ for $(i = 1, 2, ..., 10)$ for $1,000$ samples of size $20,000$. Each letter-value box plot (Hofmann et al. 2017) exhibits approximately the shape of an exponential distribution.

Following Schwarz (2008), Burridge & Taylor (2006) and Wilkinson (2017), we start our anomalous threshold calculation from a subset of the points covering 50 per cent of them with the smallest anomalous scores under the assumption that this subset contains the

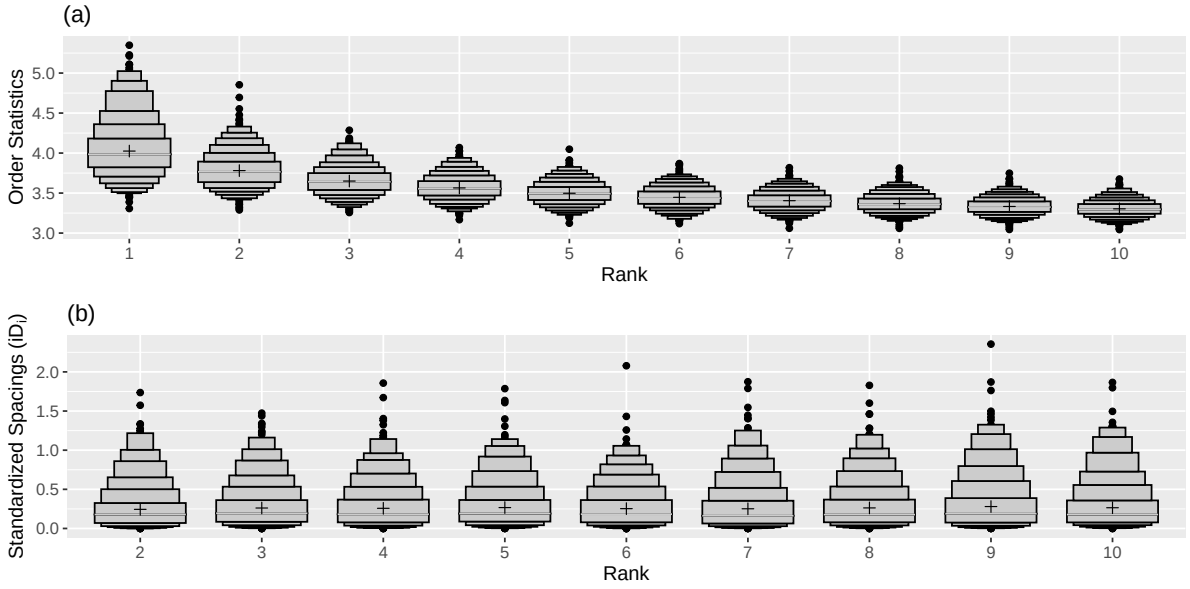


Figure 5: (a) Distribution of the descending order statistics $X_{i:n}$ and (b) distribution of the standardised spacings $iD_{i,n}$ for $i \in \{1, \ldots, 10\}$ for $1,000$ samples each containing $20,000$ random numbers from the standard normal distribution.

anomalous scores corresponding to typical data points and the remaining subset contains the scores corresponding to the possible candidates for anomalies. Following the Weissman spacing theorem, it then fits an exponential distribution to the upper tail of the outlier scores of the first subset, and then computes the upper $1-\alpha$ points of the fitted cumulative distribution function, thereby defining an anomalous threshold for the next anomalous score. Then, from the remaining subset it selects the point with the smallest anomalous score. If this anomalous score exceeds the cut-off point, it flags all the points in the remaining subset as anomalies and stops searching for anomalies. Otherwise, it declares the point as a typical point and adds it to the subset of the typical points. It then updates the cut-off point, including the latest addition. This searching algorithm continues until it finds an anomalous score that exceeds the latest cut-off point. This algorithm is known as a 'bottom-up searching' algorithm in Schwarz (2008). This threshold calculation is performed under the assumption that the distribution of k-nearest neighbours with the maximum gap is in the maximum domain of attraction of the Gumbel distribution, which covers a wide range of distributions.

4.5 Output

In stray, anomalies are measured in two scales: (1) binary classification and (2) outlier score. Under binary classification, data instances are classified either as typical or anomalous using the data-driven anomalous threshold based on the extreme value theory. This type of classification is important if the subsequent steps of the data analysis process are automated. The stray algorithm also assigns an anomalous score to each data instance to indicate the degree of outlierness of each measurement. These anomalous scores allow the user to rank and select the most serious or relevant anomalous points for root-cause analysis and taking immediate precautions. The HDoutliers algorithm (Wilkinson 2017), which provides only a binary classification, does not directly allow the user to make such a choice to direct their attention to more significant anomalous instances. Conversely, various methods proposed in the literature provide anomalous scores, but the anomalous threshold is user defined and application specific (Madsen 2018). The output produced by stray is an all-in-one solution encapsulating necessary measurements of anomalies for further actions.

5 Experiments

The HDoutliers algorithm is a powerful algorithm in the current state-of-the-art methods for detecting anomalies in high-dimensional data. The focus of the stray algorithm is to address some of the limitations of the HDoutliers algorithm that hinder its performance under certain circumstances. Here, we perform an experimental evaluation on the accuracy and computational efficiency of our stray algorithm relative to the HDoutliers algorithm. While these examples are fairly limited in number and are mostly limited to bivariate datasets, they should be viewed only as simple illustrations of the key features of the stray algorithm that outperforms the HDoutliers algorithm.

The first experiment (Figure 6) was designed to test the effect of the dimension, size of the data and the k-nearest neighbour searching method on running times of the different versions of the two algorithms: stray and HDoutliers.

The HDoutliers algorithm has two versions. The first version calculates nearest neighbour distance for each data instance and does not involve any clustering step prior to the

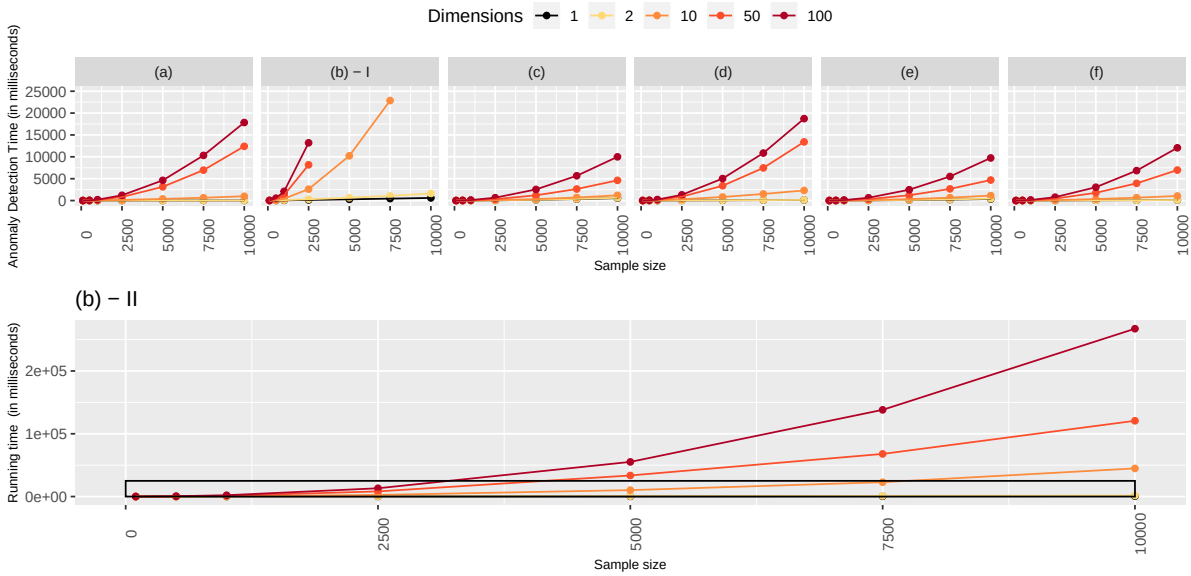


Figure 6: Scalability Performance. (a) HDoutliers algorithm without clustering step, (b-I) HDoutliers algorithm with clustering step, (c) stray algorithm with brute force nearest neighbour search using FNN R package implementation, (d) stray algorithm with kd-trees nearest neighbour search using 'FNN' R package implementation, (e) stray algorithm with brute force nearest neighbour search using 'nabor' R package implementation, (f) stray algorithm with kd-trees nearest neighbour search using 'nabor' R package implementation. For clear comparison, only a part of the measurements of the full experiment is displayed in (b-I). (b-II) presents the full version of (b-I). Black frame in (b-II) covers the plotting region of (b-I).

nearest neighbour distance calculation. This version of the algorithm (version 1 of the HDoutliers, hereafter) is recommended for small samples (n<10,000). The second version uses the Leader algorithm to form several clusters of points and then selects a representative member from each cluster. The nearest neighbour distances are then calculated only for the selected representative members. Compared with version 1 of the HDoutliers algorithm (Figure 6 (a)), version 2 with the clustering step is extremely slow for higher dimensions (>10), and the running time increases more rapidly with increasing sample size. For clear comparison between the different versions of the two algorithms (stray and HDoutliers), only a part of the measurements of the full experiment of the second version of the HD-

outliers algorithm is displayed in Figure 6 (b-I)). Figure 6 (b-II) presents the full version of Figure 6 (b-I). The additional clustering step in the second version of the HDoutliers algorithm, which is essential for detecting micro clusters, is extremely time-consuming, particularly with large samples with higher dimensions. Figure 6 (c)–(f) corresponds to the stray algorithm. In this experiment, to ascertain the influence from the k-nearest neighbour searching methods, we considered both exact (brute force) and approximate (kd-trees) nearest neighbour searching algorithms.

Many implementations of k-nearest neighbour searching algorithms are available for the R software environment. We considered the `FNN` (Beygelzimer et al. (2019), Figure 6 (c) & (d)) and `nabor` (Elseberg et al. (2012*b*); Figure 6 (e) & (f)) R packages for our comparative analysis. R package `nabor`, wraps a fast k-nearest neighbour library written in templated C++. We noticed that searching $k - (> 1)$ nearest neighbours (Figure 6 (a), in this example k is set to 10) instead of only one ($k = 1$) nearest neighbour (Figure 6 (d)) increases the running time only slightly as the number of instances is increased. The results in both Figure 6 (a) and Figure 6 (d) are based on approximate nearest neighbour distances using the kd-trees nearest neighbour searching algorithm. We observed that the kd-trees implementation in the `nabor` package (Figure 6 (f)) is much faster than the `FFN` package implementation (Figure 6 (d)). Surprisingly, as the dimension increases, the running time of the stray algorithm with kd-trees (Figure 6 (d), (f)) increases much more quickly than that of the brute force algorithm, which involves searching every possible pairing of points to detect k-nearest neighbours for each data instance (Figure 6 (c), (e)). Other studies (Kanungo et al. 2002) have also reported a similar result for algorithms based on kd-trees and its variants. This could be due to the parallelisability and memory access patterns of the two searching mechanisms. The brute force algorithm is easily parallelisable because it involves independent searching of all possible candidates for each data instance. In contrast, the kd-tree searching algorithm is naturally serial and therefore difficult to implement on parallel systems with appreciable speedup (Zhang 2017).

Following Wilkinson (2017), we evaluated the false positive rate (typical points incorrectly identified as anomalies) of the stray algorithm by running it many times on random data. The values presented in Table 1 are based on 1000 iterations and the mean values

are reported. Different versions of the two algorithms (stray and Hdoutliers) were applied on datasets where each column is randomly generated from the standardised normal distribution. In each test, the critical value, α, was set to 0.05. Compared with the HDoutliers algorithm, low false positive rates were achieved for the stray algorithm across all dimensions and sample sizes. Unlike in the HDoutliers algorithm (Unwin 2019), in stray a much smaller false detection rate was observed even for the small datasets with smaller dimensions. No difference was observed across different versions of the stray algorithm with different nearest neighbour searching mechanisms and their different implementations.

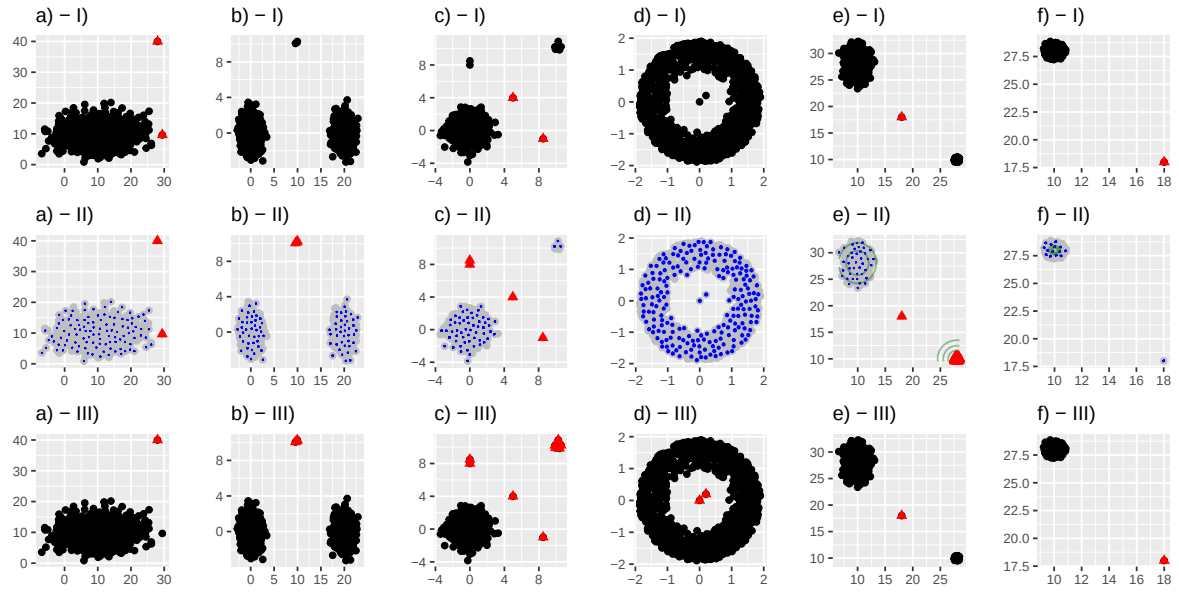


Figure 7: Algorithm performance. (a) The top panel shows the results of the HDoutliers algorithm without a clustering step. (b) The middle panel shows the results of the HDoutliers algorithm with a clustering step. The representative member selected from each cluster formed by the Leader algorithm are marked in blue colour. (c) The bottom panel shows the results of the improved algorithm with brute force k-nearest neighbour searching. The detected anomalies are marked as red triangles.

Figure 7 demonstrates how the stray algorithm outperforms the two versions of the HDoutliers algorithm under different circumstances. These limited set of examples were selected with the aim of highlighting some of the key feature of the stray algorithm:

(1) All three algorithms were able to correctly capture the anomalous point at the right-

Table 1: Performance metrics – False positive rates. The values given are based on 100 iterations and the mean values are reported. Different versions of the two algorithms (stray and Hdoutliers) are applied on datasets where each column is randomly generated from the standardised normal distribution. All the datasets are free from anomalies HDoutliers WoC: HDoutliers algorithm without clustering step; HDoutliers WC: HDoutliers algorithm with clustering step.

Method	dim	100	500	1000	2500	5000	7500	10000
HDoutliers WoC	1	0.017	0.011	0.008	0.007	0.005	0.005	0.004
HDoutliers WoC	10	0.002	0.002	0.002	0.002	0.002	0.002	0.002
HDoutliers WoC	100	0.001	0.001	0.001	0.001	0.001	0.001	0.001
HDoutliers WC	1	0.036	0.024	0.024	0.019	0.017	0.014	0.013
HDoutliers WC	10	0.006	0.006	0.006	0.005	0.005	0.005	0.005
HDoutliers WC	100	0.003	0.003	0.003	0.003	0.003	0.003	0.003
stray - brute force	1	0.006	0.003	0.002	0.002	0.002	0.001	0.001
stray - brute force	10	0.001	0.001	0.001	0.001	0.001	0.001	0.000
stray - brute force	100	0.000	0.000	0.000	0.000	0.000	0.000	0.000
stray - FNN kd-tree	1	0.006	0.003	0.002	0.002	0.002	0.001	0.001
stray - FNN kd-tree	10	0.001	0.001	0.001	0.001	0.001	0.001	0.000
stray - FNN kd-tree	100	0.000	0.000	0.000	0.000	0.000	0.000	0.000
stray - nabor brute	1	0.006	0.003	0.002	0.002	0.002	0.001	0.001
stray - nabor brute	10	0.001	0.001	0.001	0.001	0.001	0.001	0.000
stray - nabor brute	100	0.000	0.000	0.000	0.000	0.000	0.000	0.000
stray - nabor kd-tree	1	0.006	0.003	0.002	0.002	0.002	0.001	0.001
stray - nabor kd-tree	10	0.001	0.001	0.001	0.001	0.001	0.001	0.000
stray - nabor kd-tree	100	0.000	0.000	0.000	0.000	0.000	0.000	0.000

most upper corner of Figure 7 (a)- I). However, the two versions of the HDoutliers algorithm tend to generate some false positives, particularly with the small dimensions.

(2) Figure 7 (b)- III) shows its ability to deal with multimodal typical classes. The two clusters at the bottom of the graph represent two typical classes. Only the second version of the HDoutliers algorithm (Figure 7 (b)- II) that utilises the clustering step was able to detect the top-centred micro cluster that contains three anomalous data instances. However, forming small clusters prior to the distance calculation is not always helpful in detecting micro clusters.

(3) Figure 7 (c)- II) shows a situation where even the second version of the HDoutliers algorithm fails in detecting micro clusters. The Leader algorithm in the HDoutliers algorithm uses a very small ball of a fixed radius to form clusters, and therefore, it now fails to capture the five points into a single cluster and instead generates three small clusters that are very close to one another. Both versions of the HDoutliers algorithm now fail to detect the micro cluster at the rightmost upper corner, because the dataset violates one of the major requirements of isolation of anomalous points or anomalous clusters. In stray, the value of k was set to 10. One can interpret the value of k as the maximum permissible size for a micro cluster. That is, for a small cluster to be a micro cluster, the number of data points in that cluster should be less than k. Otherwise, the cluster is considered a typical cluster.

(4) Figure 7 (d)- III) demonstrates the ability of detecting inliers. The HDoutliers algorithm also has this ability of detecting inliers only when there are isolated inliers that are free from anomalous neighbours. Both versions of the HDoutliers algorithm fail to detect the two inliers since they are very close to one another and thereby jointly protect them as being anomalous.

(5) As explained in Section 3.2, Figure 7 (e)- II) shows how the clustering step of the second version of the HDoutliers algorithm can misguide the detection process and thereby increase the rate of false positives. The dense areas of the dataset are marked with density curves. Two typical clusters are visible, one at the leftmost upper corner

and the other at the rightmost bottom corner. An inlier is also present in between the two typical classes. After forming cluster through the Leader algorithm, only one representative member is selected from each cluster for the nearest neighbour distance calculation. The selected member is now isolated and earns a very high anomalous score, leading the entire typical cluster at the rightmost bottom corner with 1,000 points to be identified as anomalous. In contrast, the stray algorithm is free from these problems because it does not involve any clustering step prior to the nearest neighbour distance calculation.

(6) As explained in Section 3.2, Figure 7 (f)- II) shows how the clustering step can increase the rate of false negatives. This dataset contains one typical class that is closely compacted in substance (the leftmost upper corner) and an obvious anomaly at the rightmost bottom corner. Since the typical class is a dense cluster, only a few data points are selected from the typical class for the nearest neighbour calculation. In this example, the clustering step substantially down-samples the original dataset, leading to a huge information loss in the representation of the original dataset. The blue dots in Figure 7 (f)- II) represent the selected members from each cluster for nearest neighbour calculations. Now, the reduced sample size is not enough for a proper calculation of the anomalous threshold based on extreme value theory.

6 Usage

We applied our stray algorithm to a dataset obtained from an automated pedestrian counting system with 43 sensors in the city of Melbourne, Australia (City of Melbourne 2019, Wang 2018), to identify unusual pedestrian activities within the municipality. Identification of such unusual, critical behaviours of pedestrians at different city locations at different times of the day is important because it is a direct indication of a city's economic conditions, the related activities and the safety and convenience of the pedestrian experience (City of Melbourne 2019). It also guides and informs decision-making and planning. This case study also illustrates how the stray algorithm can be used to deal with other data structures, such as temporal data and streaming data using feature engineering.

6.1 Handling Temporal Data

For this study, we consider the hourly pedestrian counts from January 2, 2019, to August 18, 2019. For clear visual illustration, Figure 8 shows only a limited part of the study period with the pedestrian counts at 43 locations in the city of Melbourne at different times of the day. The distribution of pedestrian counts follows a negatively skewed distribution. In general, pedestrian counts on weekdays display a bimodal distribution, while pedestrian counts on weekends follow a unimodal distribution. Now, the aim is to detect days with unusual behaviours. Since this involves a large collection of multivariate time series plots, each representing a day of the study period, manual monitoring is time-consuming and unusual behaviours are difficult to locate by visual inspection.

Detecting anomalous plots from a large collection of plots requires some pre-processing. In particular, to apply the stray algorithm, we need to convert this original dataset, with a large collection of multivariate time series plots, into a high dimensional dataset. A simple approach is to use features that describe the different shapes and patterns of the multivariate time series plots. Computing features that describe meaningful shapes and patterns in a given multivariate time series plot is straightforward with scagnostics (scatterplot diagnostics) developed by Wilkinson et al. (2005). For the current study, we select nine features: outlying, skewed, clumpy, sparse, striated, convex, skinny, stringy and monotonic (Dang & Wilkinson 2014, Wilkinson et al. 2005). Once we extract these nine features from each plot, we convert our original collection of multivariate time series plots into a dataset with nine dimensions and 228 data instances covering the study period from January 2, 2019, to August 18, 2019. Figure 9 provides feature-based representation of the original collection of multivariate time series plots. Each point in this high-dimensional data space corresponds to a single multivariate time series plot (or a day) in the original collection of multivariate time series plots. Figure 10 shows the O3 plot (Overview of Outliers plot) (Unwin 2019) summarizing feature combinations on which those days are anomalies and on what groups of features have these days been identified as anomalies. There is a row for each feature combination for which anomalies are found. Two white columns separate the feature combinations and the anomalies detected. Each row of the block on the left shows which feature combination defines the row. There are 9 columns, one for each feature and

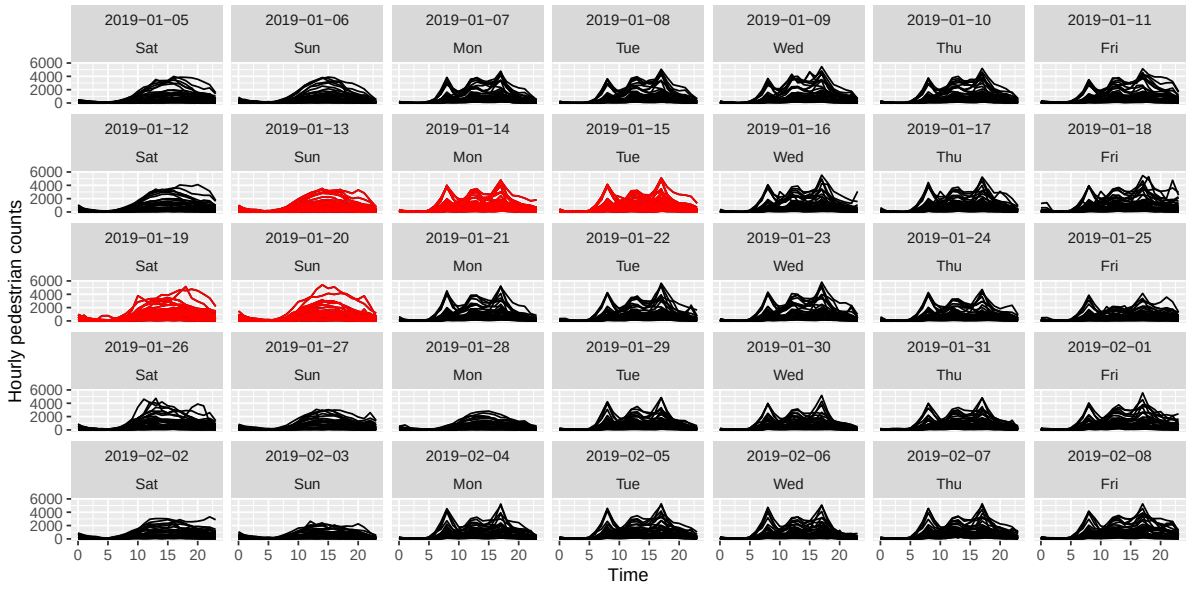


Figure 8: Collection of multivariate time series plots of hourly pedestrian counts at 43 locations in the city Melbourne, Australia, from 2 January to 8 February 2019. Anomalous days detected by the stray algorithm using scagnostics are marked in red. This covers only a small part of the study period considered (from January 2, 2019, to August 18, 2019).

a cell is gray if the feature is a part of the combination. From this analysis, 13 days were found to be anomalies in at least one of the sub feature spaces defined by different feature combinations. These anomalies are marked by red cells on the right block in Figure 10. The corresponding multivariate time series plots (or days) are marked in red in Figure 8. Visual inspection also confirms the anomalous behaviour of these individual multivariate time series plots. Most of these anomalous days display an unusual rise later in the day. Most of the anomalies in January (14, 15, 19 and 20 January 2019) cover the 2019 Australian Open, a Grand Slam tennis tournament that took place at Melbourne Park from 14 to 27 January 2019. This annual tennis tournament attracts many thousands of tennis fans from all around the worlds. Further investigations regarding 13 January 2019, reveal that there was a musical concert in Melbourne city and the unusual rise later in the day could be due to the concert participants. Similar patterns were observed with the remaining anomalies detected.

After detecting the days with anomalous pedestrian behaviours, further investigation

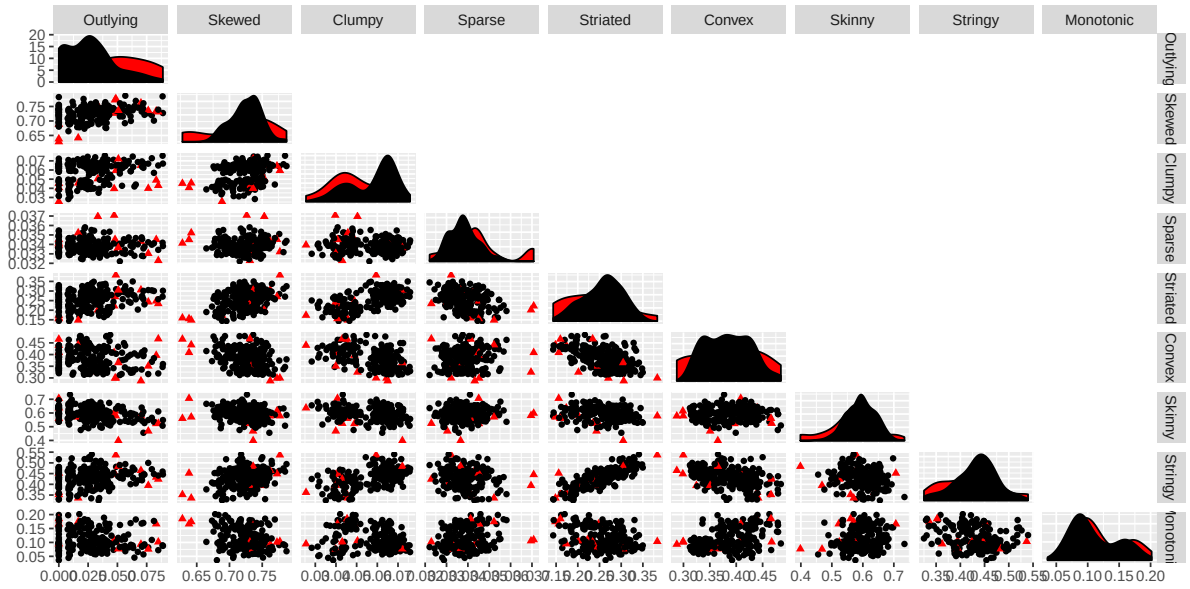


Figure 9: Feature-based representation of the collection of multivariate time series plots using scagnostics. In each plot anomalies determined by the stray algorithm in at least one of the sub feature spaces defined by different feature combinations are represented in red.

is carried out for each day to detect the locations with anomalous behaviours within the selected day. Once we focus on one day, we obtain a collection of 43 time series with hourly pedestrian counts generated from the 43 sensors located at different geographic locations in the city (Figure 11). For this analysis, we extract 11 time series features (similar to Talagala, Hyndman, Smith-Miles, Kandanaarachchi & Muñoz 2019, Hyndman et al. 2015) and convert the original collection of time series into a data space with 11 dimensions and 43 data instances (Figure 12). Now, each point in this high-dimensional space correspond to a single time series (or sensor) in Figure 11. The stray algorithm declares three points as anomalous points in this high dimensional data space. These points correspond to the sensors at Southbank, the Art centre and St Kilda Rd-Alexandra Gardens in Melbourne.

These types of findings play a critical role in making decisions about urban planning and management; to identify opportunities to improve city walkability and transport measures; to understand the impact of major events and other extreme conditions on pedestrian activity, and thereby assist in making decisions regarding security and resource requirements; and to plan and respond to emergency situations, etc.

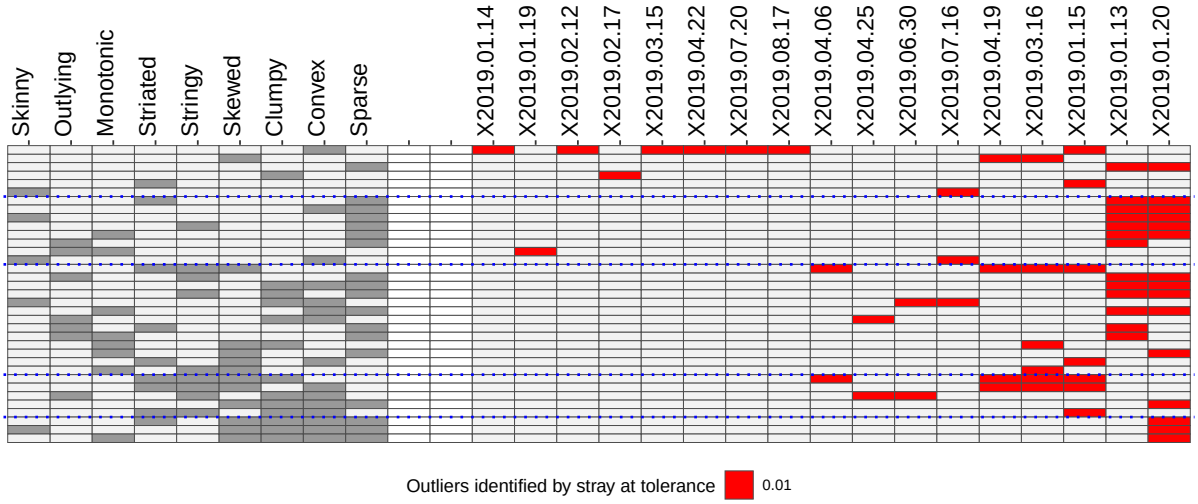


Figure 10: O3 plot of data relating to hourly pedestrian counts at 43 locations in the city Melbourne, Australia, from January 2, 2019, to August 18, 2019. Thirteen days were found to be anomalies on some combination of features. Anomalous days detected by the stray algorithm are marked in red. Two days were anomalies on several combinations, 13-01-2019 and 20-01-2019.

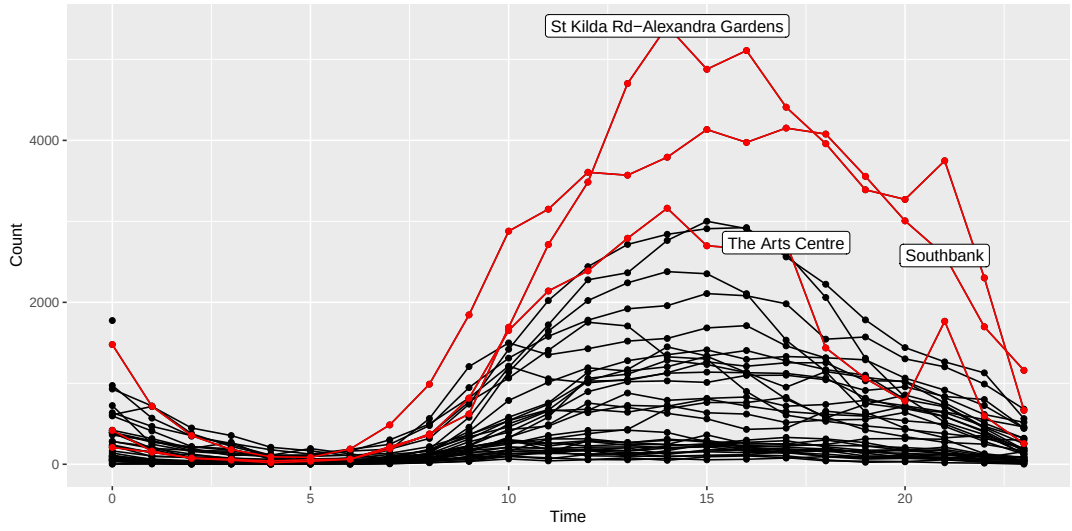


Figure 11: Multivariate time series plot of hourly counts of pedestrians measured at 43 different sensors in the city of Melbourne, on 20 Jansuary 2019. The anomalous time series detected by the stray algorithm using time series features are marked in red.

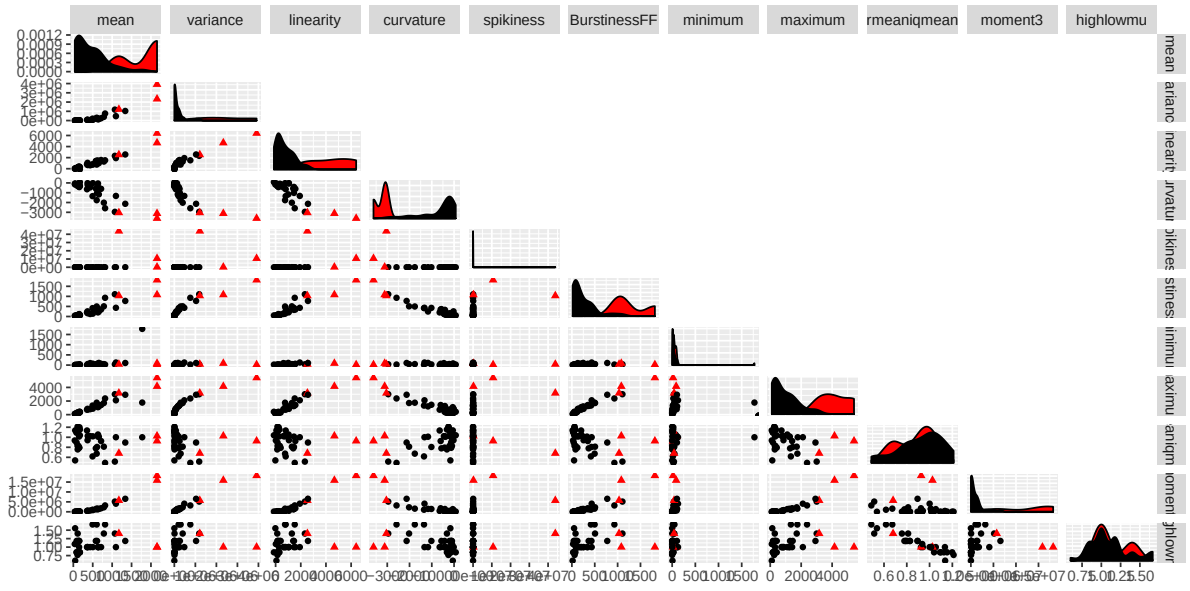


Figure 12: Feature-based representation of the collection of time series on 20 January 2019. In each plot, anomalies determined by the stray algorithm are represented in red.

6.2 Handling Streaming Data

Owing to the unsupervised nature of the stray algorithm, it can easily be extended for streaming data. A sliding window of fixed length can be used to deal with streaming data. Then, datasets in each window can be treated as a batch dataset (Talagala, Hyndman, Leigh, Mengersen & Smith-Miles 2019) and the stray algorithm can be applied to each window to detect anomalies in the datasets defined by the corresponding window.

It also can be used to identify anomalous time series within a large collection of streaming temporal data. Let $W[t, t + w]$ represent a sliding window containing n number of individual time series of length w. First, we extract m features (similar to Hyndman et al. (2015) and Talagala, Hyndman, Leigh, Mengersen & Smith-Miles (2019)) from each and every time series in this window. This step gives rise to an $n \times m$ feature matrix where each row now corresponds to a time series in the original collection of time series. Once we convert our original collection of time series into a high-dimensional dataset, we can apply the stray algorithm to identify anomalous points within this m-dimensional data space. The corresponding time series are then declared as anomalous series within the large collection of time series in the corresponding sliding window.

7 Conclusions and Further Research

The HDoutliers algorithm by Wilkinson (2017) is a powerful algorithm for detecting anomalies in high-dimensional data. However, it suffers from a few limitations that significantly hinder its ability to detect anomalies under certain situations. In this study, we propose an improved algorithm, the stray algorithm, that addresses these limitations. We define an anomaly here as an observation that deviates markedly from the majority with a large distance gap. The stray algorithm has many special features: (1) It can deal with both one dimensional and high dimensional data as it is based on distance measures. By extracting k nearest neighbour distances for each data instance, it converts any high dimensional anomaly detection problem into a one dimensional problem. (2) Since the anomalous threshold calculation is a data driven approach, the algorithm is unsupervised in nature and therefore does not require labeled training datasets. (3) Most of the existing algorithms involve a manual anomalous threshold for binary classification as anomalies or typical points. Since the stray algorithm uses an anomalous threshold based on extreme value theory, it has a valid probabilistic interpretation. (4) It deals with masking problems and detects micro clusters as it does not limit its threshold calculation only to the nearest neighbour distances and instead uses k nearest neighbour distances. (5) Since it uses fast nearest neighbour searching mechanisms it can be easily extended to streaming data using sliding windows. (6) Owing to the use of k nearest neighbour distances it can deal with multimodal distributions. (7) In addition to a label, the stray algorithm also assigns an anomalous score to each data instance to indicate the degree of outlierness of each measurement. (8) Owing to the use of k nearest neighbour distances it also detect inliers, which is overlooked in most past research. We also demonstrate how the stray algorithm can assist in detecting anomalies present in other data structures using feature engineering.

While the HDoutliers algorithm is powerful, we have provided several classes of counterexamples in this paper where the structural properties of the data did not enable HDoutliers to detect certain types of outliers. We demonstrated on these counterexamples that the stray algorithm outperforms HDoutliers, in terms of both accuracy and computational time. It is certainly common practice to evaluate the strength of an algorithm using collections of test problems with various challenging properties. However, we acknowledge that

these counterexamples are not diverse and challenging enough to enable us to comment about the unique strengths and weaknesses of these two algorithms, nor to generalise our findings to conclude that stray is always the superior algorithm. This study should be viewed as an attempt to simulate further investigation on the HDoutliers algorithm and its successors, with the ultimate goal to achieve further improvements across the entire problem space defined by various high-dimensional datasets. An important open research problem is therefore to assess the effectiveness of these algorithms across the the broadest possible problem space defined by different datasets with diverse properties (Kang et al. 2017). It is an interesting question to explore the impact of other classes of problems with various structural properties affect the performance of the stray algorithm and where its weaknesses might lie. This kind of instance space analysis (Smith-Miles et al. 2014) will enable further insights into improved algorithm design.

Anomaly detection problems commonly appear in many applications in different application domains. Therefore, it is hoped that different people with different knowledge levels will use the stray algorithm for many different purposes. Therefore, we expect future studies to develop interactive data visualisation tools that can enable exploring anomalies using a combination of graphical and numerical methods.

Supplementary Materials

Data and scripts: Datasets and R code to reproduce all figures in this article (main.R and R package `stray` (Talagala, Hyndman & Smith-Miles 2019)).

R package `stray`: The `stray` package consists of the implementation of the stray algorithm as described in this 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/priditalal/stray`.

R-packages: Each of the R packages used in this article (`ggplot2` (Wickham 2016), `dplyr` (Wickham et al. 2019), `tidyr` (Wickham & Henry 2019), `HDoutliers` (Fraley 2018), `lvplot`(Wickham & Hofmann 2016) 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. Further, we thank Sevvandi Kandanaarachchi and Mario A. Muñoz for joining the discussions during the initial stage of the project.

References

Abuzaid, A., Hussin, A. & Mohamed, I. (2013), 'Detection of outliers in simple circular regression models using the mean circular error statistic', *Journal of Statistical Computation and Simulation* **83**(2), 269–277.

Aggarwal, C. C. (2017), *Outlier analysis*, second edition. edn, Cham, Switzerland : Springer.

Aggarwal, C. C. & Yu, P. S. (2008), Outlier detection with uncertain data, *in* 'Proceedings of the 2008 SIAM International Conference on Data Mining', SIAM, pp. 483–493.

Ben-Gal, I. (2005), Outlier detection, *in* 'Data mining and knowledge discovery handbook', Springer, pp. 131–146.

Beygelzimer, A., Kakadet, S., Langford, J., Arya, S., Mount, D. & Li, S. (2019), *FNN: Fast Nearest Neighbor Search Algorithms and Applications*. R package version 1.1.3.
URL: *https://CRAN.R-project.org/package=FNN*

Breunig, M. M., Kriegel, H.-P., Ng, R. T. & Sander, J. (2000), Lof: identifying density-based local outliers, *in* 'ACM Sigmod Record', Vol. 29, ACM, pp. 93–104.

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

Campos, G. O., Zimek, A., Sander, J., Campello, R. J., Micenková, B., Schubert, E., Assent, I. & Houle, M. E. (2016), 'On the evaluation of unsupervised outlier detection:

measures, datasets, and an empirical study', *Data Mining and Knowledge Discovery* **30**(4), 891–927.

Cao, N., Shi, C., Lin, S., Lu, J., Lin, Y.-R. & Lin, C.-Y. (2015), 'Targetvue: Visual analysis of anomalous user behaviors in online communication systems', *IEEE Transactions on Visualization and Computer Graphics* **22**(1), 280–289.

Chandola, V., Banerjee, A. & Kumar, V. (2009), 'Anomaly detection: A survey', *ACM Computing Surveys* **41**(3), 1–58.

City of Melbourne (2019), *Pedestrian Volume in Melbourne*. Last accessed 2019-07-23.
URL: *http://www.pedestrian.melbourne.vic.gov.au*

Clifton, D. A., Hugueny, S. & Tarassenko, L. (2011), 'Novelty detection with multivariate extreme value statistics', *Journal of Signal Processing Systems* **65**(3), 371–389.

Dang, T. N. & Wilkinson, L. (2014), 'Transforming scagnostics to reveal hidden features', *IEEE Transactions on Visualization and Computer Graphics* **20**(12), 1624–1632.

Elseberg, J., Magnenat, S., Siegwart, R. & Nüchter, A. (2012*a*), 'Comparison of nearest-neighbor-search strategies and implementations for efficient shape registration', *Journal of Software Engineering for Robotics* **3**(1), 2–12.

Elseberg, J., Magnenat, S., Siegwart, R. & Nüchter, A. (2012*b*), 'Comparison of nearest-neighbor-search strategies and implementations for efficient shape registration', *Journal of Software Engineering for Robotics (JOSER)* **3**(1), 2–12.

Embrechts, P., Klüppelberg, C. & Mikosch, T. (2013), *Modelling Extremal Events: for Insurance and Finance*, Stochastic Modelling and Applied Probability, Springer Berlin Heidelberg.

Fraley, C. (2018), *HDoutliers: Leland Wilkinson's Algorithm for Detecting Multidimensional Outliers*. R package version 1.0.
URL: *https://CRAN.R-project.org/package=HDoutliers*

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.

Gao, J., Hu, W., Zhang, Z. M., Zhang, X. & Wu, O. (2011), Rkof: robust kernel-based local outlier detection, *in* 'Pacific-Asia Conference on Knowledge Discovery and Data Mining', Springer, pp. 270–283.

Goldstein, M. & Uchida, S. (2016), 'A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data', *PlOS ONE* **11**(4), e0152173.

Grubbs, F. E. (1969), 'Procedures for detecting outlying observations in samples', *Technometrics* **11**(1), 1–21.

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

Hartigan, J. A. & Hartigan, J. (1975), *Clustering Algorithms*, Vol. 209, Wiley New York.

Hodge, V. & Austin, J. (2004), 'A survey of outlier detection methodologies', *Artificial Intelligence Review* **22**(2), 85–126.

Hofmann, H., Wickham, H. & Kafadar, K. (2017), 'Value plots: Boxplots for large data', *Journal of Computational and Graphical Statistics* **26**(3), 469–477.

Hyndman, R. J. (1996), 'Computing and graphing highest density regions', *The American Statistician* **50**(2), 120–126.

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

Jin, W., Tung, A. K., Han, J. & Wang, W. (2006), Ranking outliers using symmetric neighborhood relationship, *in* 'Pacific-Asia Conference on Knowledge Discovery and Data Mining', Springer, pp. 577–593.

Jouan-Rimbaud, D., Bouveresse, E., Massart, D. & De Noord, O. (1999), 'Detection of prediction outliers and inliers in multivariate calibration', *Analytica Chimica Acta* **388**(3), 283–301.

Kandanaarachchi, S., Munoz, M. A., Hyndman, R. J., Smith-Miles, K. et al. (2018), On normalization and algorithm selection for unsupervised outlier detection, 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.

Kanungo, T., Mount, D. M., Netanyahu, N. S., Piatko, C. D., Silverman, R. & Wu, A. Y. (2002), 'An efficient k-means clustering algorithm: Analysis and implementation', *IEEE Transactions on Pattern Analysis & Machine Intelligence* (7), 881–892.

Leigh, C., Alsibai, O., Hyndman, R. J., Kandanaarachchi, S., King, O. C., McGree, J. M., Neelamraju, C., Strauss, J., Talagala, P. D., Turner, R. D., Mengersen, K. & Peterson, E. E. (2019), 'A framework for automated anomaly detection in high frequency water-quality data from in situ sensors', *Science of the Total Environment* **664**, 885–898.

Liu, S., Maljovec, D., Wang, B., Bremer, P.-T. & Pascucci, V. (2016), 'Visualizing high-dimensional data: Advances in the past decade', *IEEE Transactions on Visualization and Computer Graphics* **23**(3), 1249–1268.

Madsen, J. H. (2018), *DDoutlier: Distance and Density-Based Outlier Detection*. R package version 0.1.0.
URL: *https://CRAN.R-project.org/package=DDoutlier*

Novotny, M. & Hauser, H. (2006), 'Outlier-preserving focus+ context visualization in parallel coordinates', *IEEE Transactions on Visualization and Computer Graphics* **12**(5), 893–900.

Pimentel, M. A., Clifton, D. A., Clifton, L. & Tarassenko, L. (2014), 'A review of novelty detection', *Signal Processing* **99**, 215–249.

R Core Team (2019), *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria.
URL: *https://www.R-project.org/*

Sabahi, F. & Movaghar, A. (2008), Intrusion detection: A survey, *in* '3rd International Conference on Systems and Networks Communications-ICSNC'08.', IEEE, pp. 23–26.

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.

Shahid, N., Naqvi, I. H. & Qaisar, S. B. (2015), 'Characteristics and classification of outlier detection techniques for wireless sensor networks in harsh environments: a survey', *Artificial Intelligence Review* **43**(2), 193–228.

Smith-Miles, K., Baatar, D., Wreford, B. & Lewis, R. (2014), 'Towards objective measures of algorithm performance across instance space', *Computers & Operations Research* **45**, 12–24.

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., Leigh, C., Mengersen, K. & Smith-Miles, K. (2019), 'A feature-based procedure for detecting technical outliers in water-quality data from in situ sensors', *Water Resources Research* (accepted).

Talagala, P. D., Hyndman, R. J. & Smith-Miles, K. (2019), *stray: Anomaly Detection in High Dimensional and Temporal Data*. R package version 0.1.0.9000.

Talagala, P. D., Hyndman, R. J., Smith-Miles, K., Kandanaarachchi, S. & Muñoz, M. A. (2019), 'Anomaly detection in streaming nonstationary temporal data', *Journal of Computational and Graphical Statistics* pp. 1–21.

Tang, J., Chen, Z., Fu, A. W.-C. & Cheung, D. W. (2002), Enhancing effectiveness of outlier detections for low density patterns, *in* 'Pacific-Asia Conference on Knowledge Discovery and Data Mining', Springer, pp. 535–548.

Unwin, A. (2019), 'Multivariate outliers and the o3 plot', *Journal of Computational and Graphical Statistics* pp. 1–11.

Wang, E. (2018), *rwalkr: API to Melbourne Pedestrian Data*. R package version 0.4.0.
URL: *https://CRAN.R-project.org/package=rwalkr*

Weissman, I. (1978), 'Estimation of parameters and large quantiles based on the k largest observations', *Journal of the American Statistical Association* **73**(364), 812–815.

Wickham, H. (2016), *ggplot2: Elegant Graphics for Data Analysis*, Springer-Verlag New York.
URL: *https://ggplot2.tidyverse.org*

Wickham, H., François, R., Henry, L. & Müller, K. (2019), *dplyr: A Grammar of Data Manipulation*. R package version 0.8.3.
URL: *https://CRAN.R-project.org/package=dplyr*

Wickham, H. & Henry, L. (2019), *tidyr: Tidy Messy Data*. R package version 1.0.0.
URL: *https://CRAN.R-project.org/package=tidyr*

Wickham, H. & Hofmann, H. (2016), *lvplot: Letter Value 'Boxplots'*. R package version 0.2.0.
URL: *https://CRAN.R-project.org/package=lvplot*

Wilkinson, L. (2017), 'Visualizing big data outliers through distributed aggregation', *IEEE Transactions on Visualization and Computer Graphics* **24**(1), 256–266.

Wilkinson, L., Anand, A. & Grossman, R. (2005), Graph-theoretic scagnostics, *in* 'IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.', IEEE, pp. 157–164.

Williams, K. T. (2016), Local parametric density-based outlier detection and ensemble learning with applications to malware detection, PhD thesis, The University of Texas at San Antonio.

Zhang, R. (2017), 'Performance of kd-tree vs brute-force nearest neighbor search on gpu?', Computational Science Stack Exchange. URL:https://scicomp.stackexchange.com/q/26873 (version: 2017-05-13).

Zhang, W., Wu, J. & Yu, J. (2010), An improved method of outlier detection based on frequent pattern, *in* 'Information Engineering (ICIE), 2010 WASE International Conference on', Vol. 2, IEEE, pp. 3–6.

Chapter 3

Anomaly Detection in Streaming Non-stationary Temporal Data

This article is published in the *Journal of Computational and Graphical Statistics*.

Journal of Computational and Graphical Statistics

ISSN: 1061-8600 (Print) 1537-2715 (Online) Journal homepage: https://www.tandfonline.com/loi/ucgs20

Anomaly Detection in Streaming Nonstationary Temporal Data

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

To cite this article: Priyanga Dilini Talagala, Rob J. Hyndman, Kate Smith-Miles, Sevvandi Kandanaarachchi & Mario A. Muñoz (2019): Anomaly Detection in Streaming Nonstationary Temporal Data, Journal of Computational and Graphical Statistics, DOI: 10.1080/10618600.2019.1617160

To link to this article: https://doi.org/10.1080/10618600.2019.1617160

JOURNAL OF COMPUTATIONAL AND GRAPHICAL STATISTICS
https://doi.org/10.1080/10618600.2019.1617160

Anomaly Detection in Streaming Nonstationary Temporal Data

Priyanga Dilini Talagala[a,b], Rob J. Hyndman[a,b], Kate Smith-Miles[b,c], Sevvandi Kandanaarachchi[a,b], and Mario A. Muñoz[b,c]

[a]Department of Econometrics and Business Statistics, Monash University, Clayton, Victoria, Australia; [b]ARC Centre of Excellence for Mathematics and Statistical Frontiers (ACEMS), Australia; [c]School of Mathematics and Statistics, University of Melbourne, Parkville, Victoria, Australia

ABSTRACT

This article proposes a framework that provides early detection of anomalous series within a large collection of nonstationary 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 nonstationarity 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 online supplementary materials.

ARTICLE HISTORY
Received February 2018
Revised April 2019

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 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 and 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 article makes three fundamental contributions to anomaly detection in streaming nonstationary 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 nonstationary environments (also known as "concept drift" in the machine learning literature). We reduce the collection of time series to a two-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 resampling 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 and Sui 2009), electrical short circuits (Krohn, MacDougall, and Mendez 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 fiber may indicate intrusions. Similarly, sensor cables can monitor temperature profiles along gas and oil pipelines, allowing the detection of leakages (Krohn, MacDougall, and 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 nonstationary 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 nonstationary environments. Some simulations illustrating the

CONTACT Priyanga Talagala dilini.talagala@monash.edu Department of Econometrics and Business Statistics, Monash University, Clayton, VIC 3800, Australia.
Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/r/JCGS.
Supplementary materials for this article are available online. Please go to www.tanfonline.com/r/JCGS.

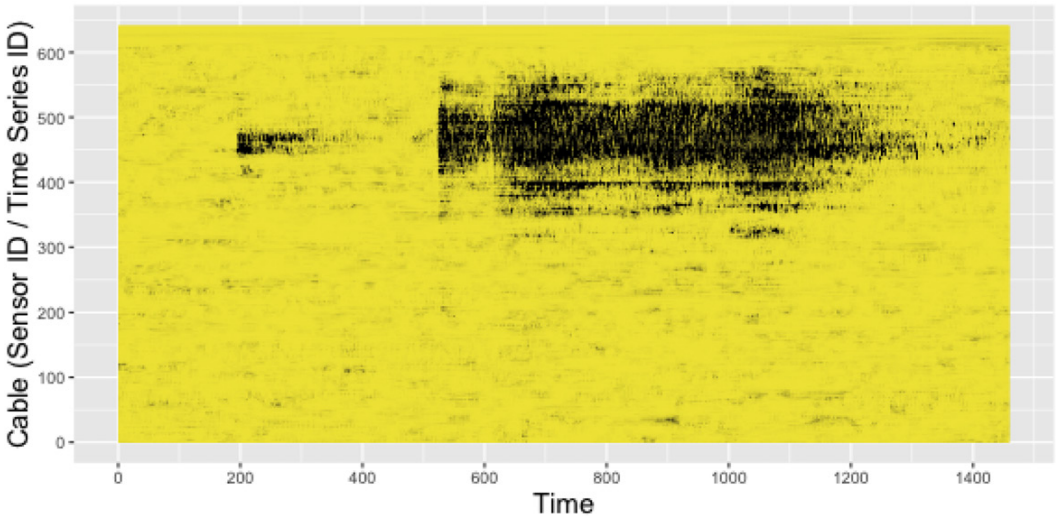


Figure 1. Multivariate time series plot of a dataset obtained using a fiber optic cable. Axis "Cable" represents individual points of the sensor cable. There are 640 time series each with 1459 time points. Yellow corresponds to low values and black to high values. The black region near the upper endpoint of the cable (around 350–500) 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.

method are presented in Section 4. An application of the proposed framework is given in Section 5. Section 6 concludes the article.

2. Background

2.1. Types of Anomalies in Temporal Data

The problems of anomaly detection for temporal data are 3-fold: (a) the detection of contextual anomalies within a given series; (b) the detection of anomalous subsequences 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 and Capretz 2015). Burridge and 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 and Huzurbazar 2002) or multivariate problems (Riani, Atkinson, and Cerioli 2009; Galeano, Peña, and Tsay 2006; Peña and 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 and Taylor (2006).

The final setting, the detection of anomalous series within a collection of series, is the primary focus of this article. 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, and Laptev (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, Broderick, and Wand 2014). With batch processing, as in Hyndman, Wang, and Laptev (2015) and Wilkinson (2018), it is assumed that the entire dataset 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, Lechner, and Simiu 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

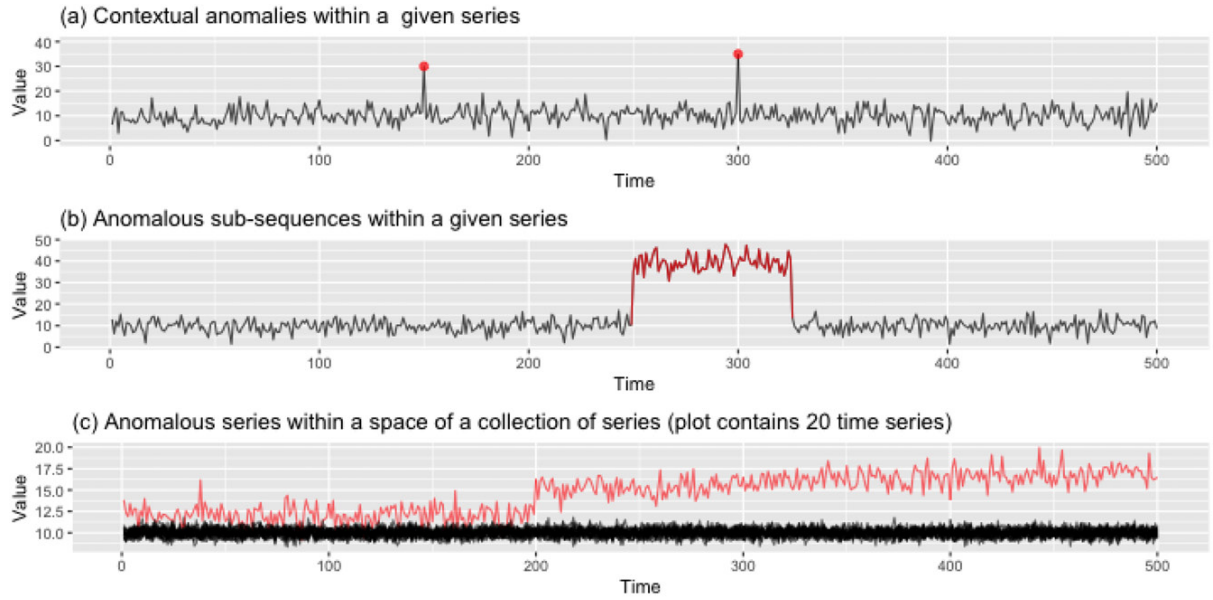


Figure 2. Different types of anomalies in temporal data. In each plot anomalies are represented by red color and black color is corresponding to the typical behavior.

$X_{\max} = \max(\boldsymbol{X})$ 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, Hugueny, and Tarassenko 2011). The Fisher–Tippett theorem (Fisher and 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, Klüppelberg, and Mikosch (2013, p. 121); the notation has been changed for consistency.

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 nondegenerate 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, Klüppelberg, and Mikosch (2013) discussed 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, Klüppelberg, and 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 EVT

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 and Taylor (2006), Schwarz (2008), and Wilkinson (2018) provided 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 and Rodríguez (2003), on which the method of Burridge and Taylor (2006) was based, mentioned the possibility of using EVT and nonparametric estimates of tail behavior, but did not provide any detailed discussion. Sundaram et al. (2009), Clifton, Hugueny, and Tarassenko (2011), and Hugueny (2013) provided 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.

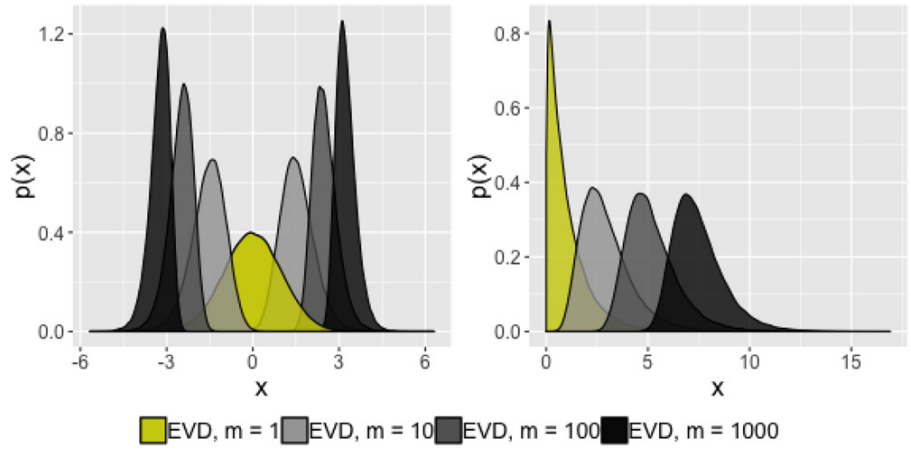


Figure 3. Empirical distributions of 10^6 minima and maxima for the standard Gaussian distribution (left), and of maxima for the standard exponential distribution (right). (Reproduced from Hugueny 2013, p. 87.)

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, Hugueny, and Tarassenko (2011) proposed a numerical method for selecting a threshold for identifying anomalous points when $m \geq 1$. In their "Ψ transform method," Clifton, Hugueny, and Tarassenko (2011) defined 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; that is, $\arg\min_{x \in X}[f(x)]$.

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, Hugueny, and 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 *online* (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, Wang, and Laptev (2015) and Wilkinson (2018).

First, the method proposed by Hyndman, Wang, and 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 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 4(b), 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 4(b)). 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, Hugueny, and Tarassenko 2011).

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, Wang, and Laptev 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 nonstationarity of the time series (Kang, Hyndman, and Li 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, Wang, and Laptev (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, Wang, and Laptev 2015 and Kang, Hyndman, and Smith-Miles 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, Hyndman, and Smith-Miles 2017).
5. Estimate the probability density of this *2D PC space* using kernel density estimation with a bivariate Gaussian kernel (similar to Luca et al. 2014 and 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, Hugueny, and Tarassenko 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, that is, 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, Hugueny, and Tarassenko 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 and Worden 2012).

As recommended by Jin and 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 (Online 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.

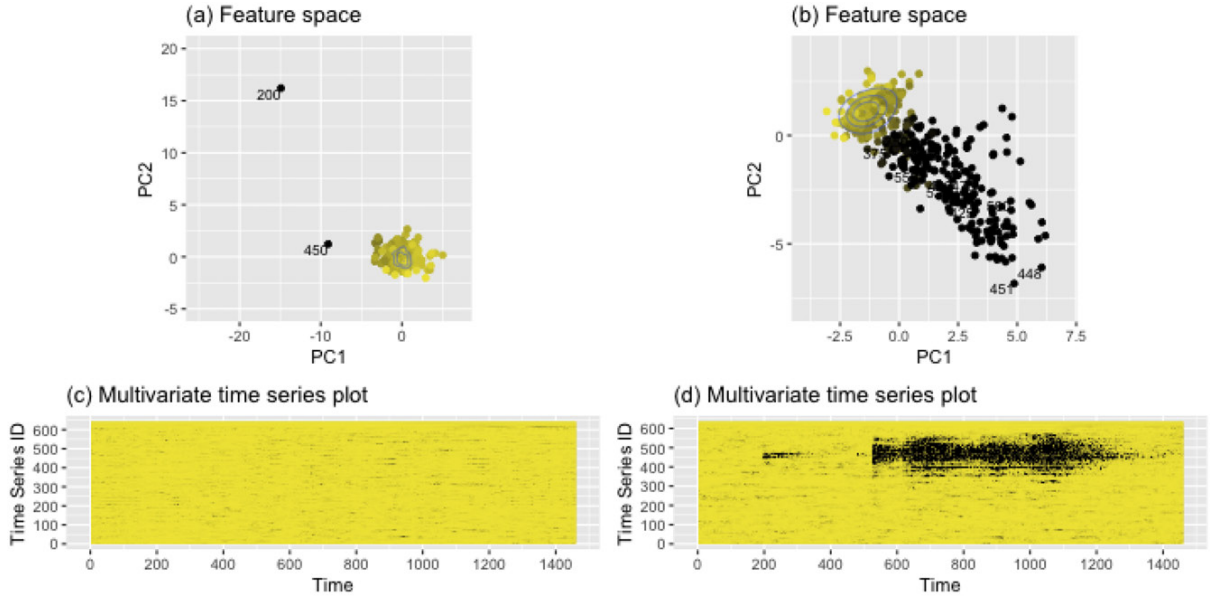


Figure 4. Left panel corresponding to a collection of time series obtained via independent sensors. Right panel corresponding to a collection of time series obtained via sensors that are not independent to one another. Black: high values; yellow: low values. Black dots/lines/shapes are corresponding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

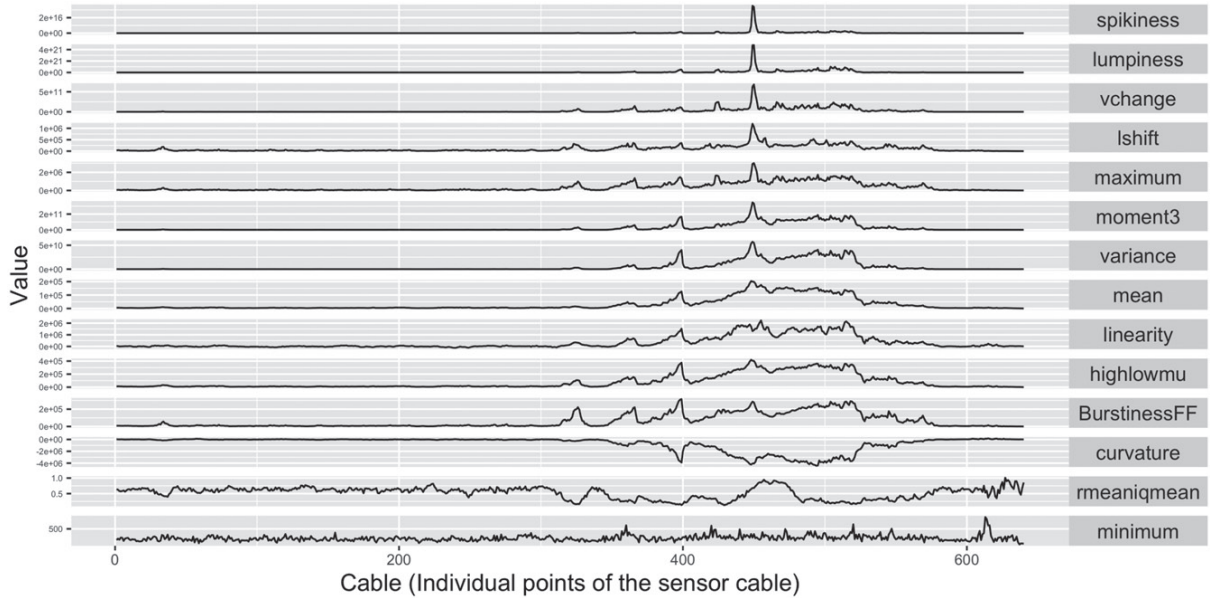


Figure 5. Feature based representation of the time series in Figure 1. There are 640 time series ($m = 640$). Each plot is corresponding to a feature type extracted from the 640 time series ($k = 14$). Almost all the features have captured the unusual event near the right endpoint of the cable (around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

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 online phase for every new time window that is generated by the current time point, t.

3.2. Handling Nonstationary 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, Banerjee, and Kumar 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 nonstationarity, 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 nonstationary data distribution, it is necessary to regularly update the model to account for changes in the data distribution. In the econometrics literature, these nonstationary environments are sometimes classified as either "structural breaks" or "time-varying" evolutionary change (Rapach and 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, Sebastião, and Rodrigues (2013), there are two approaches that can be used to adapt models 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 nonstationary (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 and 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) declared 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 and Rückert (2009), we propose an informed approach for early detection of nonstationarity 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 $W[t-w,t]_{\text{norm}} (\subseteq W[t-w,t])$ 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 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 7–9 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, Hall, and Titterington 1994). However, these computationally intensive resampling 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, and Schauer (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, Hall, and Titterington (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 resampling 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 $W[t-w,t]_{\text{norm}}$. 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 (Figures 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 and 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 and 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 Figures 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 $W[250,400]$. 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.

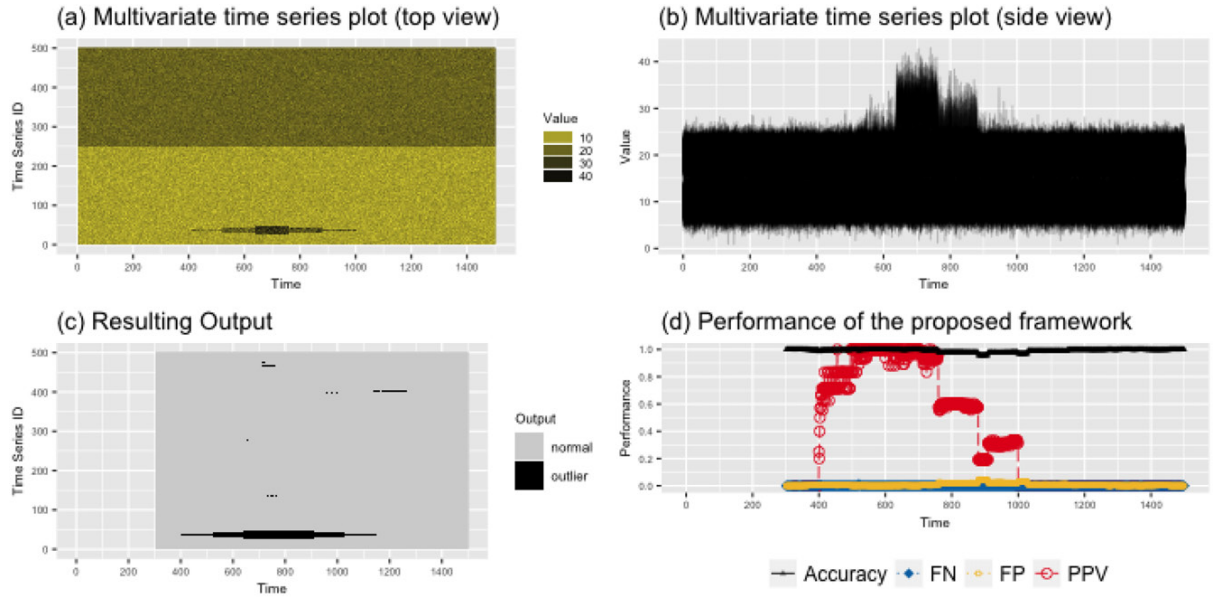


Figure 6. Multimodal typical classes but no nonstationarity. Sliding window length = 150 time points. To initiate the algorithm, $W[1, 150]$ is considered as a representative sample of the typical behavior. (a) Multivariate time series plot of the collection of time series ($m = 300$). The upper half 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 $F_2^{\epsilon} = 0.999$. (d) Performance of the proposed framework (without any adjustments to nonstationary 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 $t = 520$).

One-class support vector machine (OCSVM) is a commonly used method in anomaly detection research (Ma and Perkins 2003; Mahadevan and Shah 2009; Rajasegarar et al. 2010). Raskutti and Kowalczyk (2004) and Zhuang and 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, and 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 Nonstationary 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 nonstationarity: *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 7–10, 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, nonstationary 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 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 7–9, 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.

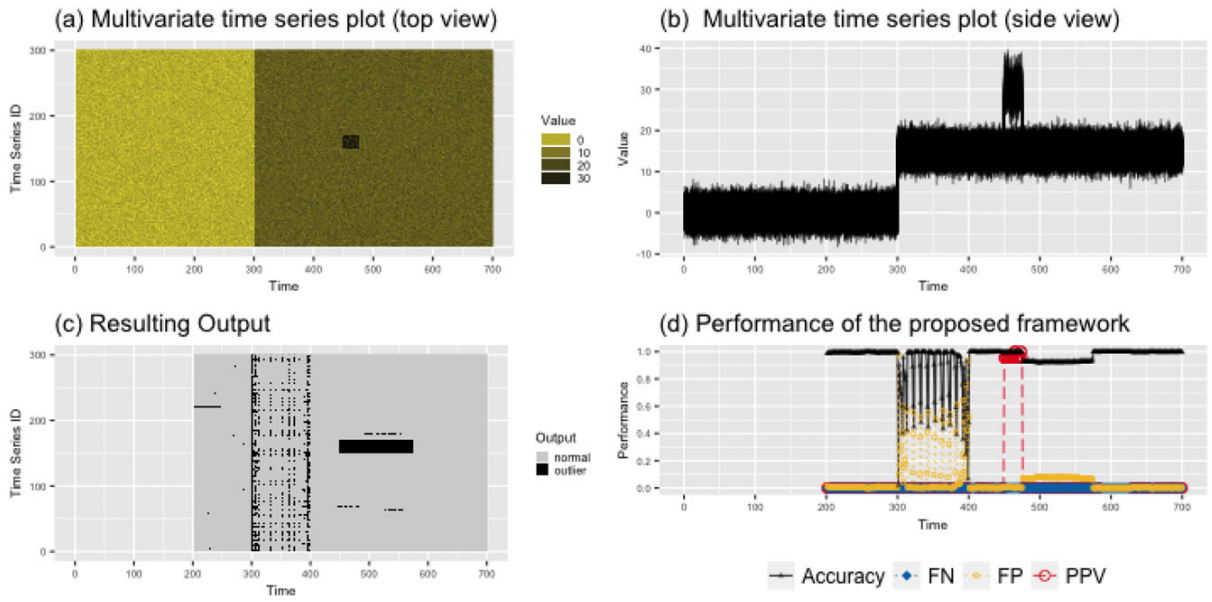


Figure 7. "Sudden" nonstationarity. (a) Multivariate time series plot of the collection of time series ($m = 300$). "Sudden" nonstationarity starting from t =300. (b) Multivariate time series plot (side view of panel (a)). (c) The output produced by the sliding window approach. In the test phase the 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 $t = 450$).

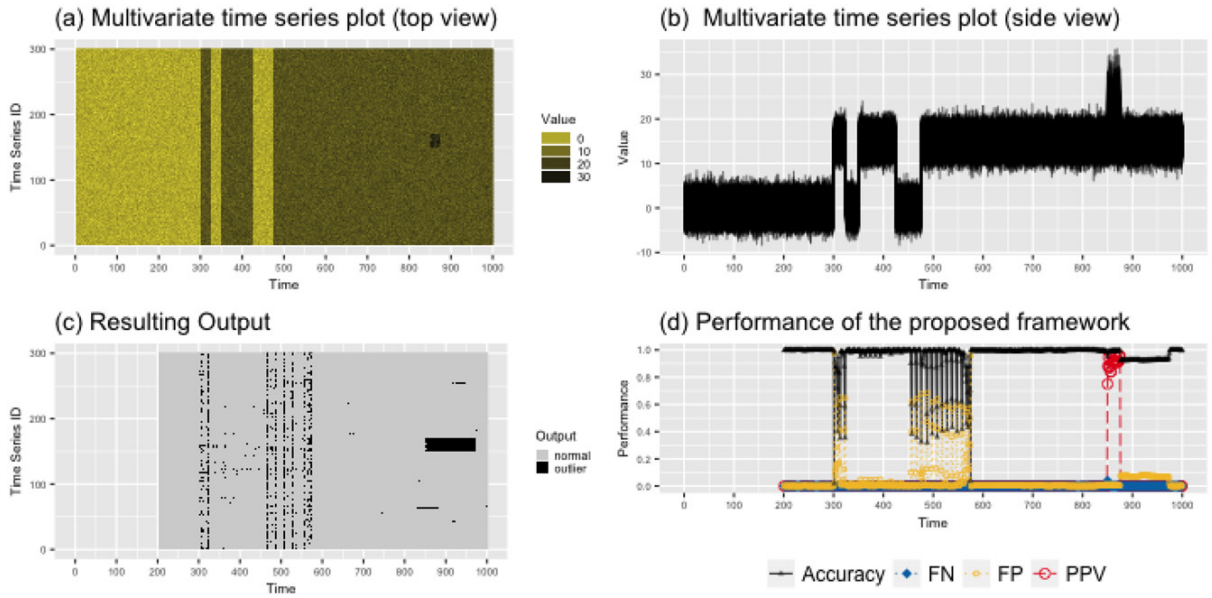


Figure 8. "Gradual" nonstationarity. (a) Multivariate time series plot of the collection of time series ($m = 300$). "Gradual" nonstationarity starting from t =300. (b) Multivariate time series plot (side view of panel (a)). (c) The output produced by the sliding window approach. In the test phase the 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 $t = 850$).

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 1–3), 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 $H_0 : f_{t_0} = f_{t_t}$ 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

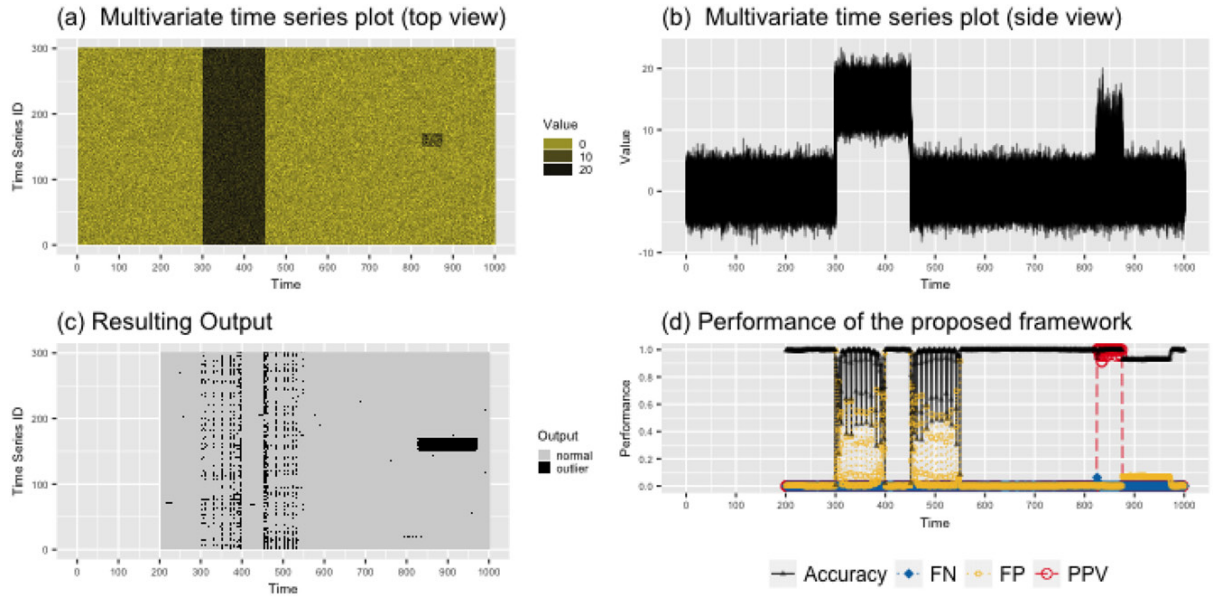


Figure 9. "Reoccurring" type nonstationarity. (a) Multivariate time series plot of the collection of time series ($m = 300$). "Reoccurring" type nonstationarity starting from $t = 300$. (b) Multivariate time series plot (side view of panel (a)). (c) The output produced by the sliding window approach. 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 $t = 825$).

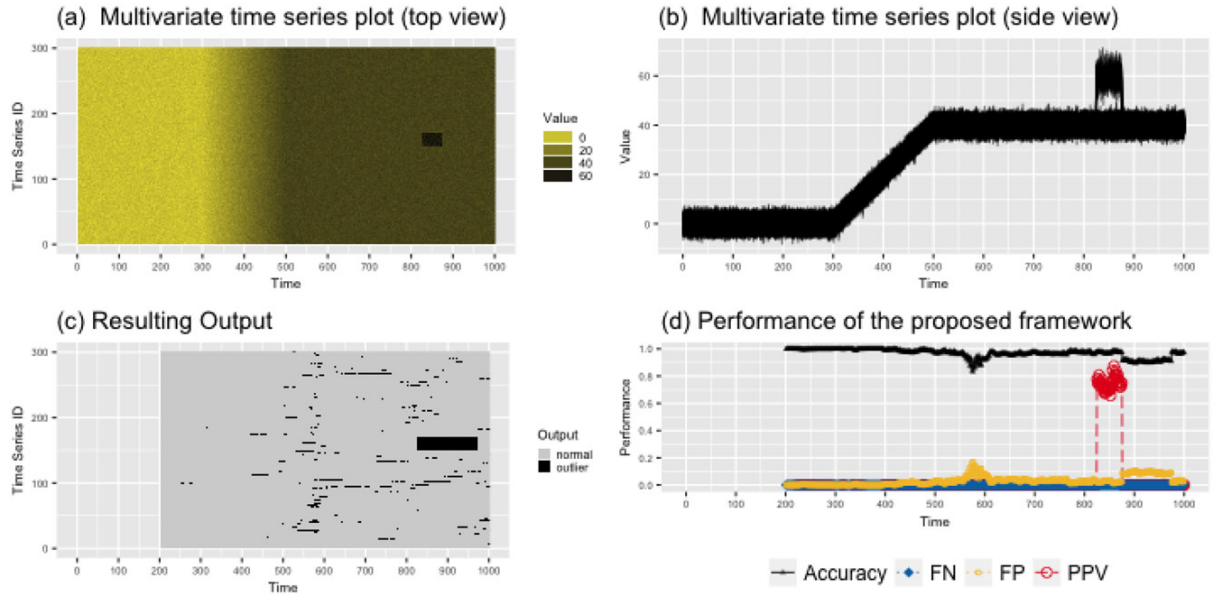


Figure 10. "Incremental" nonstationarity. (a) Multivariate time series plot of the collection of time series ($m = 300$). "Incremental" nonstationarity starting from $t = 300$. (b) Multivariate time series plot (side view of panel (a)). (c) The output produced by the sliding window approach. In the test 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 $t = 201$).

(i.e., claiming the occurrence of nonstationarity 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 nonstationarity 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 1–3 to datasets obtained using fiber optic sensor cables attached to a system. (Since the

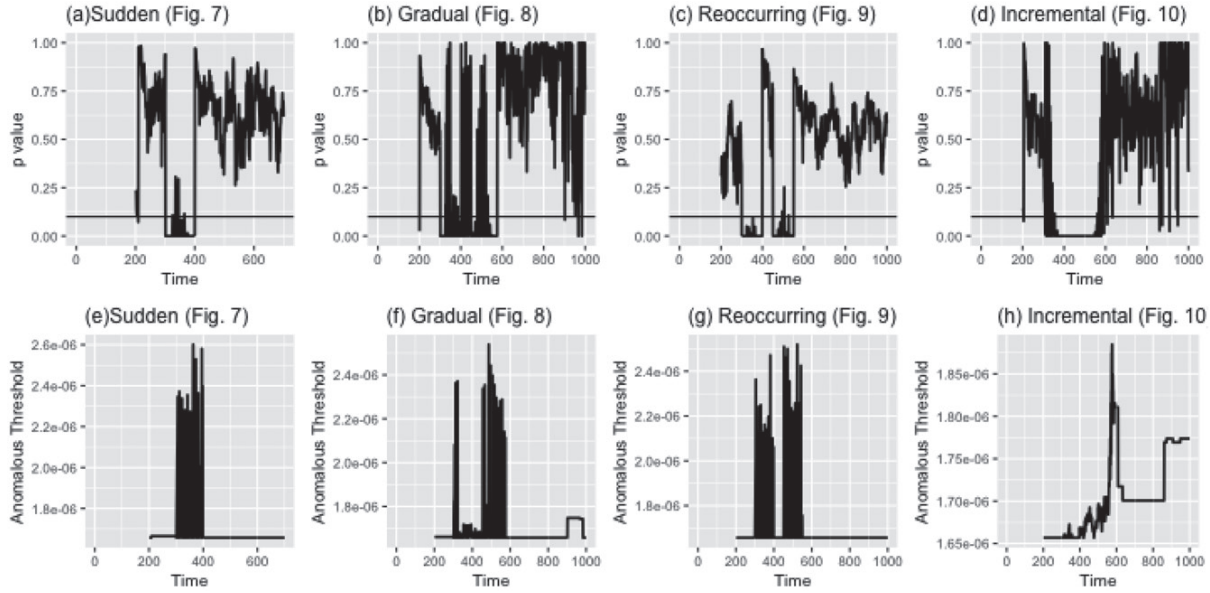


Figure 11. Detection of nonstationarity. Top panel: *p*-value for the hypothesis test $H_0 : f_{t_0} = f_{t_t}$. In these examples the significance level is set to 0.1 and is marked by the horizontal line in each plot. Bottom panel: Anomalous threshold.

data contain commercially sensitive information, this article 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.

As explained in Section 3, our proposed algorithm requires a representative sample of the typical behavior of each of these datasets 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) and (b)) and $W[1, 50]$ for the third example (Figure 12(c)) as the representative sample of the typical behavior 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 nonstationary data distributions allows the model to adjust to the system's typical behavior over time. Figure 13 gives the corresponding *p*-values for the hypothesis test $H_0 : f_{t_0} = f_{t_t}$ 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 1–3. 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) and (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) and (f)). Therefore, our proposed framework is particularly well suited for the applications described in Section 1, which generate large collections of time series.

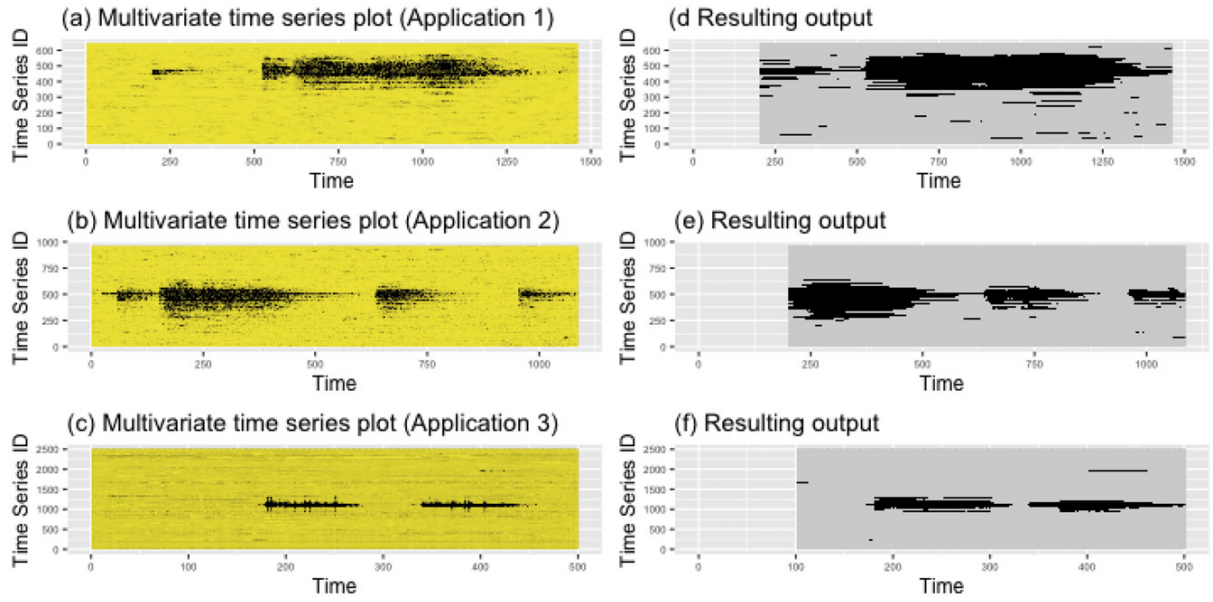


Figure 12. Application (Application 1: m = 640, Application 2: m = 1000, Application 3: m = 2500). Left panel: black: high values; yellow: low values; black shapes are corresponding to anomalous events. Right panel: black: outliers; gray: typical behavior.

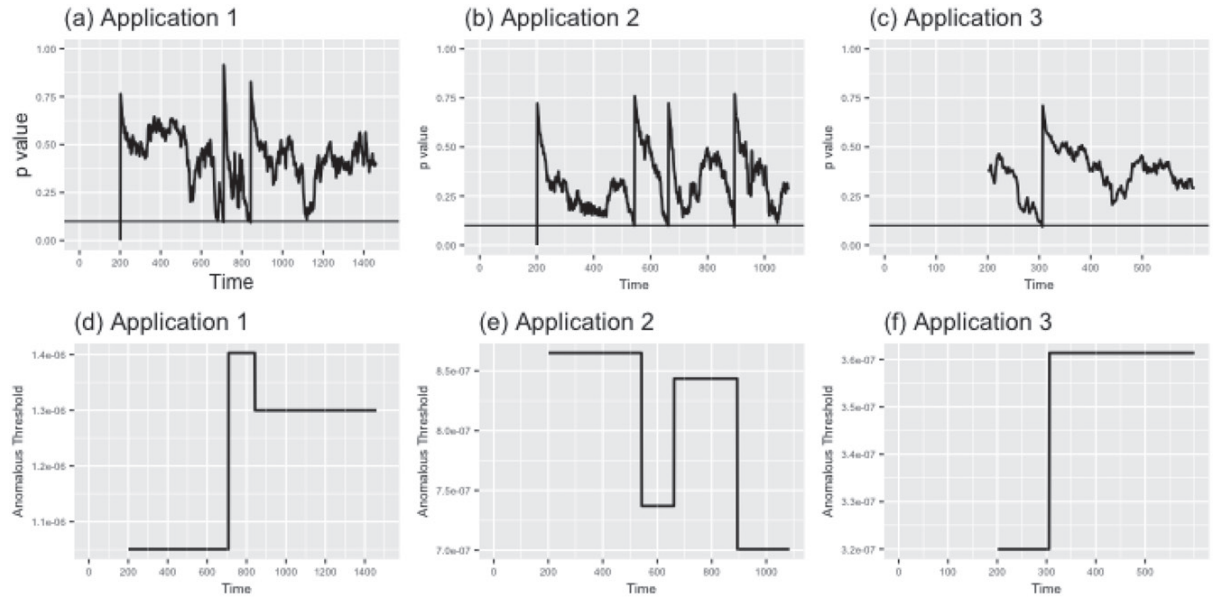


Figure 13. Detection of nonstationarity. Top panel: *p*-value for the hypothesis test $f_{to} = f_{tt}$. In these examples the significance level is set to 0.1 and is marked by the horizontal line in each plot. Bottom panel: Anomalous threshold.

6. Conclusions and Further Work

This article 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 nonstationary 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 1–3) can work well in the presence of nonstationary 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 non-stationarity also depends on the kernel density estimates, and we may not reject stationarity when m is small. Log-

spline bivariate density estimation (Kooperberg and Stone 1991) and local likelihood density estimation (Loader 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 and 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, and Vanrumste 2014; Luca, Clifton, and Vanrumste 2016; Luca et al. 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 non-stationarity. 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.

Supplementary Materials

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

R package oddstream: The oddstream package (Talagala, Hyndman, and Smith-Miles 2018) consists of the implementation of Algorithms 1–3 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 and Wickham 2017; *tidyr*, Wickham and Henry 2017; *reshape*, Wickham 2007) are available online (URLs are provided in the bibliography).

Funding

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., and 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, 41–54. [7]

Bilen, C., and Huzurbazar, S. (2002), "Wavelet-Based Detection of Outliers in Time Series," *Journal of Computational and Graphical Statistics*, 11, 311–327. [2]

Burridge, P., and Taylor, A. M. R. (2006), "Additive Outlier Detection via Extreme-Value Theory." *Journal of Time Series Analysis*, 27, 685–701. [2,3]

Catalano, A., Bruno, F. A., Pisco, M., Cutolo, A., and Cusano, A. (2014), "An Intrusion Detection System for the Protection of Railway Assets Using Fiber Bragg Grating Sensors," *Sensors*, 14, 18268–18285. [1]

Chandola, V., Banerjee, A., and Kumar, V. (2009), "Anomaly Detection: A Survey," *ACM Computing Surveys (CSUR)*, 41, 15. [6]

Clifton, D. A., Clifton, L., Hugueny, S., and Tarassenko, L. (2014), "Extending the Generalised Pareto Distribution for Novelty Detection in High-Dimensional Spaces," *Journal of Signal Processing Systems*, 74, 323–339. [14]

Clifton, D. A., Clifton, L., Hugueny, S., Wong, D., and Tarassenko, L. (2013), "An Extreme Function Theory for Novelty Detection," *IEEE Journal of Selected Topics in Signal Processing*, 7, 28–37. [14]

Clifton, D. A., Hugueny, S., and Tarassenko, L. (2011), "Novelty Detection With Multivariate Extreme Value Statistics," *Journal of Signal Processing Systems*, 65, 371–389. [3,4,5]

Cuppens, K., Karsmakers, P., Van de Vel, A., Bonroy, B., Milosevic, M., Luca, S., Croonenborghs, T., Ceulemans, B., Lagae, L., Van Huffel, S., and Vanrumste B. (2014), "Accelerometry-Based Home Monitoring for Detection of Nocturnal Hypermotor Seizures Based on Novelty Detection," *IEEE Journal of Biomedical and Health Informatics*, 18, 1026–1033. [5,6]

Dries, A., and Rückert, U. (2009), "Adaptive Concept Drift Detection," *Statistical Analysis and Data Mining*, 2, 311–327. [7]

Duong, T., Goud, B., and Schauer, K. (2012), "Closed-Form Density-Based Framework for Automatic Detection of Cellular Morphology Changes," *Proceedings of the National Academy of Sciences of the United States of America*, 109, 8382–8387. [7]

Embrechts, P., Klüppelberg, C., and Mikosch, T. (2013), *Modelling Extremal Events: for Insurance and Finance* (Stochastic Modelling and Applied Probability), Berlin: Springer. Available at *https://books.google.com.au/books?id=BXOI2pICfJUC* [3]

Faria, E. R., Gonçalves, I. J., de Carvalho, A. C., and Gama, J. (2016), "Novelty Detection in Data Streams," *Artificial Intelligence Review*, 45, 235–269. [2,4,7]

Farrar, C. R., and Worden, K. (2012), *Structural Health Monitoring: A Machine Learning Perspective*, Hoboken, NJ: Wiley. [5]

Fisher, R. A., and 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: Cambridge University Press, pp. 180–190. [3]

Fulcher, B. D. (2012), "Highly Comparative Time-Series Analysis," PhD thesis, University of Oxford. [5]

Galambos, J., Lechner, J., and Simiu, E. (2013), *Extreme Value Theory and Applications: Proceedings of the Conference on Extreme Value Theory and Applications*, (Vol. 1), Gaithersburg, MD: Springer US. Available at *https://books.google.com.au/books?id=XMPkBwAAQBAJ* [2]

Galeano, P., Peña, D., and Tsay, R. S. (2006), "Outlier Detection in Multivariate Time Series by Projection Pursuit," *Journal of the American Statistical Association*, 101, 654–669. [2]

Gama, J., Sebastião, R., and Rodrigues, P. P. (2013), "On Evaluating Stream Learning Algorithms,", *Machine Learning*, 90, 317–346. [7]

Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., and Bouchachia, A. (2014), "A Survey on Concept Drift Adaptation," *ACM Computing Surveys (CSUR)*, 46, 44. [7,9]

Gupta, M., Gao, J., Aggarwal, C., and Han, J. (2014), "Outlier Detection for Temporal Data: A Survey," *IEEE Transactions on Knowledge and Data Engineering*, 26, 2250–2267. [2]

Hayes, M., and Capretz, M. (2015), "'Contextual Anomaly Detection Framework for Big Sensor Data," *Journal of Big Data*, 2, 2. [2]

Hossin, M., and Sulaiman, M. (2015), "A Review on Evaluation Metrics for Data Classification Evaluations," *International Journal of Data Mining & Knowledge Management Process*, 5, 1. [8]

Hugueny, S. (2013), "Novelty Detection With Extreme Value Theory in Vital-Sign Monitoring," PhD thesis, University of Oxford. [3,4,5]

Hyndman, R. J., Wang, E., and Laptev, N. (2015), "Large-Scale Unusual Time Series Detection," in *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, IEEE, pp. 1616–1619. [2,4,5]

Jiang, Q., and Sui, Q. (2009), "Technological Study on Distributed Fiber Sensor Monitoring of High Voltage Power Cable in Seafloor," in *IEEE International Conference on Automation and Logistics, 2009. ICAL'09*, IEEE, pp. 1154–1157. [1]

Jin, R., and Agrawal, G. (2007), "Frequent Pattern Mining in Data Streams," in *Data Streams*, Boston, MA: Springer, pp. 61–84. [5]

Kang, Y., Hyndman, R. J., and Li, F. (2018), "Efficient Generation of Time Series With Diverse and Controllable Characteristics," Technical Report, Monash University, Department of Econometrics and Business Statistics. [5]

Kang, Y., Hyndman, R. J., and Smith-Miles, K. (2017), "Visualising Forecasting Algorithm Performance Using Time Series Instance Spaces," *International Journal of Forecasting*, 33, 345–358. [5]

Kooperberg, C., and Stone, C. J. (1991), "A Study of Logspline Density Estimation," *Computational Statistics & Data Analysis*, 12, 327–347. [14]

Krohn, D. A., MacDougall, T., and Mendez, A. (2000), *Fiber Optic Sensors: Fundamentals and Applications*, Triangle Park, NC: ISA. [1]

Lavin, A., and Ahmad, S. (2015), "Evaluating Real-Time Anomaly Detection Algorithms—The Numenta Anomaly Benchmark, in *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, IEEE, pp. 38–44. [1]

Loader, C. R. (1996), "Local Likelihood Density Estimation," *The Annals of Statistics*, 24, 1602–1618. [14]

Luca, S., Clifton, D. A., and Vanrumste, B. (2016), "One-Class Classification of Point Patterns of Extremes," *The Journal of Machine Learning Research*, 17, 6581–6601. [14]

Luca, S. E., Pimentel, M. A., Watkinson, P. J., and Clifton, D. A. (2018), "Point Process Models for Novelty Detection on Spatial Point Patterns and Their Extremes," *Computational Statistics & Data Analysis*, 125, 86–103. [14]

Luca, S., Karsmakers, P., Cuppens, K., Croonenborghs, T., Van de Vel, A., Ceulemans, B., Lagae, L., Van Huffel, S., and Vanrumste, B. (2014), "Detecting Rare Events Using Extreme Value Statistics Applied to Epileptic Convulsions in Children," *Artificial Intelligence in Medicine*, 60, 89–96. [5]

Luca, S., Karsmakers, P., and Vanrumste, B. (2014), "Anomaly Detection Using the Poisson Process Limit for Extremes, in *2014 IEEE International Conference on Data Mining (ICDM)*, IEEE, pp. 370–379. [9,14]

Luts, J., Broderick, T., and Wand, M. P. (2014), "Real-Time Semiparametric Regression," *Journal of Computational and Graphical Statistics*, 23, 589–615. [2]

Ma, J., and Perkins, S. (2003), "Time-Series Novelty Detection Using One-Class Support Vector Machines," in *Proceedings of the International Joint Conference on Neural Networks, 2003* (Vol. 3), IEEE, pp. 1741–1745. [9]

Mahadevan, S., and Shah, S. L. (2009), "Fault Detection and Diagnosis in Process Data Using One-Class Support Vector Machines," *Journal of Process Control*, 19, 1627–1639. [9]

Marshall, A. W., and Olkin, I. (2007), *Life Distributions* (Vol. 13), Berlin: Springer. [14]

Moshtaghi, M., Bezdek, J. C., Havens, T. C., Leckie, C., Karunasekera, S., Rajasegarar, S., and Palaniswami, M. (2014), "Streaming Analysis in Wireless Sensor Networks," *Wireless Communications and Mobile Computing*, 14, 905–921. [6,7]

Müller, K., and Wickham, H. (2017), *tibble: Simple Data Frames*, R package version 1.4.1. Available at *https://CRAN.R-project.org/package=tibble* [14]

Nikles, M. (2009), "Long-Distance Fiber Optic Sensing Solutions for Pipeline Leakage, Intrusion, and Ground Movement Detection," *SPIE: Fiber Optic Sensors and Applications VI*, 7316, 731602. [1]

O'Reilly, C., Gluhak, A., Imran, M. A., and Rajasegarar, S. (2014), "Anomaly Detection in Wireless Sensor Networks in a Non-Stationary Environment," *IEEE Communications Surveys & Tutorials*, 16, 1413--1432. [6,7]

Peña, D., and Prieto, F. J. (2001), "Multivariate Outlier Detection and Robust Covariance Matrix Estimation," *Technometrics*, 43, 286–310. [2]

Perron, P., and Rodríguez, G. (2003), "Searching for Additive Outliers in Nonstationary Time Series," *Journal of Time Series Analysis*, 24, 193–220. [3]

Rajasegarar, S., Leckie, C., Bezdek, J. C., and 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, 518–533. [9]

Ranawana, R., and Palade, V. (2006), "Optimized Precision—A New Measure for Classifier Performance Evaluation," in *IEEE Congress on Evolutionary Computation, 2006. CEC 2006*, IEEE, pp. 2254–2261. [8]

Rapach, D. E., and Strauss, J. K. (2008), "Structural Breaks and Garch Models of Exchange Rate Volatility," *Journal of Applied Econometrics*, 23, 65–90. [7]

Raskutti, B., and Kowalczyk, A. (2004), "Extreme Re-Balancing for SVMS: A Case Study," *ACM Sigkdd Explorations Newsletter*, 6, 60--69. [9]

Riani, M., Atkinson, A. C., and Cerioli, A. (2009), "Finding an Unknown Number of Multivariate Outliers," *Journal of the Royal Statistical Society*, Series B, 71, 447–466. [2]

Rodríguez, J. J., and 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)*, Berlin: Springer, pp. 520–529. [7]

Schwarz, K. T. (2008), *Wind Dispersion of Carbon Dioxide Leaking From Underground Sequestration, and Outlier Detection in Eddy Covariance Data Using Extreme Value Theory*, Berkeley, CA: University of California, Berkeley. [2,3]

Sundaram, S., Strachan, I. G. D., Clifton, D. A., Tarassenko, L., and 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*. [3]

Talagala, P. D., Hyndman, R. J., and Smith-Miles, K. (2018), *oddstream: Outlier Detection in Data Streams*, R package version 0.5.0. Available at *https://github.com/pridiltal/oddstream* [14]

Wickham, H. (2007), "Reshaping Data With the Reshape Package," *Journal of Statistical Software*, 21, 1–20. Available at *http://www.jstatsoft.org/v21/i12/paper* [14]

——— (2009), *ggplot2: Elegant Graphics for Data Analysis*, New York: Springer-Verlag. Available at *http://ggplot2.org* [14]

Wickham, H., Francois, R., Henry, L., and Müller, K. (2017), *dplyr: A Grammar of Data Manipulation*, R package version 0.7.4. Available at *https://CRAN.R-project.org/package=dplyr* [14]

Wickham, H., and Henry, L. (2017), *tidyr: Easily Tidy Data with "spread()" and "gather()" Functions*, R package version 0.7.2. Available at *https://CRAN.R-project.org/package=tidyr* [14]

Wilkinson, L. (2018), "Visualizing Big Data Outliers Through Distributed Aggregation," *IEEE Transactions on Visualization and Computer Graphics*, 24, 256–266. [2,3,4]

Yoon, S., Ye, W., Heidemann, J., Littlefield, B., and Shahabi, C. (2011), "Swats: Wireless Sensor Networks for Steamflood and Waterflood Pipeline Monitoring," *IEEE Network*, 25, 50–56. [1]

Zhang, Y., Meratnia, N., and Havinga, P. J. (2010), "Ensuring High Sensor Data Quality Through Use of Online Outlier Detection Techniques," *International Journal of Sensor Networks*, 7, 141–151. [7]

Zhuang, L., and Dai, H. (2006), "Parameter Estimation of One-Class SVM on Imbalance Text Classification," in *Conference of the Canadian Society for Computational Studies of Intelligence*, Berlin: Springer, pp. 538–549. [9]

Chapter 4

A Feature-Based Procedure for Detecting Technical Outliers in Water-Quality Data from *in situ* Sensors

This article is published in the *Water Resources Research*. The work is based on the collaborative research project carried out with the Queensland University of Technology and the Queensland Department of Environment and Science, Great Barrier Reef Catchment Loads Monitoring Program from April to July 2018.

Water Resources Research

RESEARCH ARTICLE
10.1029/2019WR024906

A Feature-Based Procedure for Detecting Technical Outliers in Water-Quality Data From In Situ Sensors

Priyanga Dilini Talagala[1,2], Rob J. Hyndman[1,2], Catherine Leigh[1,3,4], Kerrie Mengersen[1,4], and Kate Smith-Miles[1,5]

[1]ARC Centre of Excellence for Mathematics and Statistical Frontiers (ACEMS), Melbourne, Victoria, Australia, [2]Department of Econometrics and Business Statistics, Monash University, Clayton, Victoria, Australia, [3]Institute for Future Environments, Science and Engineering Faculty, Queensland University of Technology, Brisbane, Queensland, Australia, [4]School of Mathematical Sciences, Science and Engineering Faculty, Queensland University of Technology, Brisbane, Australia, [5]School of Mathematics and Statistics, University of Melbourne, Parkville, Victoria, Australia

Key Points:
- Feature-based procedure starts by applying different statistical transformations to data to highlight outliers in high-dimensional space
- Density- and distance-based unsupervised outlier scoring techniques were applied to detect outliers due to technical issues with the sensors
- An approach based on extreme value theory was then used to calculate outlier thresholds

Supporting Information:
- Supporting Information S1

Correspondence to:
P. D. Talagala,
dilini.talagala@monash.edu

Citation:
Talagala, P. D., Hyndman, R. J., Leigh, C., Mengersen, K., & Smith-Miles, K. (2019). A feature-based procedure for detecting technical outliers in water-quality data from in situ sensors. *Water Resources Research*, *55*. https://doi.org/10.1029/2019WR024906

Received 1 FEB 2019
Accepted 16 SEP 2019
Accepted article online 12 OCT 2019

Abstract Outliers due to technical errors in water-quality data from in situ sensors can reduce data quality and have a direct impact on inference drawn from subsequent data analysis. However, outlier detection through manual monitoring is infeasible given the volume and velocity of data the sensors produce. Here we introduce an automated procedure, named oddwater, that provides early detection of outliers in water-quality data from in situ sensors caused by technical issues. Our oddwater procedure is used to first identify the data features that differentiate outlying instances from typical behaviors. Then, statistical transformations are applied to make the outlying instances stand out in a transformed data space. Unsupervised outlier scoring techniques are applied to the transformed data space, and an approach based on extreme value theory is used to calculate a threshold for each potential outlier. Using two data sets obtained from in situ sensors in rivers flowing into the Great Barrier Reef lagoon, Australia, we show that oddwater successfully identifies outliers involving abrupt changes in turbidity, conductivity, and river level, including sudden spikes, sudden isolated drops, and level shifts, while maintaining very low false detection rates. We have implemented this oddwater procedure in the open source R package `oddwater`.

1. Introduction

Water-quality monitoring traditionally relies on water samples collected manually. The samples are then analyzed within laboratories to determine the water-quality variables of interest. This type of rigorous laboratory analysis of field-collected samples is crucial in making natural resources management decisions that affect human welfare and environmental conditions. However, with the rapid advances in hardware technology, the use of in situ water-quality sensors positioned at different geographic sites is becoming an increasingly common practice used to acquire real-time measurements of environmental and water-quality variables. Though only a subset of the required water-quality variables can be measured by these sensors, they have several advantages. Their ability to collect large quantities of data and to archive historic records allows for deeper analysis of water-quality variables to improve understanding about field conditions and water-quality processes (Glasgow et al., 2004). Near-real-time monitoring also allows operators to identify and respond to potential issues quickly and thus manage the operations efficiently. Further, the use of in situ sensors can greatly reduce the labor involved in field sampling and laboratory analysis.

Water-quality sensors are exposed to changing environments and extreme weather conditions and thus are prone to errors, including failure. Automated detection of outliers in water-quality data from in situ sensors has therefore captured the attention of many researchers both in the ecology and data science communities (Archer et al., 2003; Hill et al., 2009; Koch & McKenna, 2010; McKenna et al., 2007; Raciti et al., 2012). This problem of outlier detection in water-quality data from in situ sensors can be divided into two subtopics according to their focus: (1) identifying errors in the data due to issues unrelated to water events per se, such as technical aberrations, that make the data unreliable and untrustworthy and (2) identifying real events (e.g., rare but sudden spikes in turbidity associated with rare but sudden high-flow events). Both problems are equally important when making natural resource management decisions that affect human welfare and

environmental conditions. Problem 1 can also be considered as a data preprocessing phase before addressing Problem 2.

In this work we focus on Problem 1, that is, detecting unusual measurements caused by technical errors that make data unreliable and untrustworthy and affect performance of any subsequent data analysis under Problem 2. According to Yu (2012), the degree of confidence in the sensor data is one of the main requirements for a properly defined environmental analysis procedure. For instance, researchers and policy makers are unable to use water-quality data containing technical outliers with confidence for decision making and reporting purposes because erroneous conclusions regarding the quality of the water being monitored could ensue, leading, for example, to inappropriate or unnecessary water treatment, land management, or warning alerts to the public (Kotamäki et al., 2009; Rangeti et al., 2015). Missing values and corrupted data can also have an adverse impact on water-quality model building and calibration processes (Archer et al., 2003). Early detection of these technical outliers will limit the use of corrupted data for subsequent analysis. For instance, it will limit the use of corrupted data in real-time forecasting and online applications such as online drinking water-quality monitoring and early warning systems (Storey et al., 2011), predicting algal bloom outbreaks leading to fish kill events and potential human health impacts, forecasting water level and currents, and so on (Archer et al., 2003; Glasgow et al., 2004; Hill & Minsker, 2006). However, because data arrive near continuously at high speed in large quantities, manual monitoring is highly unlikely to be able to capture all the errors. These issues have therefore increased the importance of developing automated methods for early detection of outliers in water-quality data from in situ sensors (Hill et al., 2009).

Different statistical approaches are available to detect outliers in water-quality data from in situ sensors. For example, Hill and Minsker (2006) addressed the problem of outlier detection in environmental sensors using regression-based time series models. In this work they addressed the scenario as a univariate problem. Their prediction models are based on four data-driven methods: naive, clustering, perceptron, and Artificial Neural Networks (ANNs). Measurements that fell outside the bounds of an established prediction interval were declared as outliers. They also considered two strategies: anomaly detection and anomaly detection and mitigation for the detection process. Anomaly detection and mitigation replaces detected outliers with the predicted value prior to the next predictions, whereas anomaly detection simply uses the previous measurements without making any alteration to the detected outliers. These types of data-driven methods develop models using sets of training examples containing a feature set and a target output. Later, Hill et al. (2009) addressed the problem by developing three automated anomaly detection methods using dynamic Bayesian networks and showed that dynamic Bayesian network-based detectors, using either robust Kalman filtering or Rao-Blackwellized particle filtering, outperformed that of Kalman filtering.

Another common approach for detecting outliers in environmental sensor data is based on residuals (the differences between predicted and actual values). Due to the ability of ANNs to model a wide range of complex nonlinear phenomena, Moatar et al. (1999) used ANN techniques to detect anomalies such as abnormal values, discontinuities, and drifts in pH readings. After developing the pH model, the Student t test and the cumulative Page-Hinkley test were applied to detect changes in the mean of the residuals to detect measurement error occurring over short periods of time. The work was later expanded to a multivariate scenario with some additional water-quality variables including dissolved oxygen, electrical conductivity, pH, and temperature (Moatar et al., 2001). Their proposed algorithm used both deterministic and stochastic approaches for the model building process. Observed data were then compared with the model forecasts using a set of classical statistical tests to detect outliers, demonstrating the effectiveness and advantages of the multimodel approach. Later, Archer et al. (2003) proposed a method to detect failures in the water-quality sensors due to biofouling based on a sequential likelihood ratio test. Their method also had the ability to provide estimates of biofouling onset time, which was useful for the subsequent step of outlier correction.

A common feature of all of the above methods is that they are usually employed in a supervised or semisupervised context and thus require training data prelabeled with known outliers or data that are free from the anomalous features of interest. In many cases, however, not all the possible outliers are known in advance and can arise spontaneously as new outlying behaviors during the test phase. In such situations, supervised methods may fail to detect those outliers. Semisupervised methods are also unsuitable for certain applications due to the unavailability of training data containing only typical instances that are free from outliers (Goldstein & Uchida, 2016). The data sets that we consider in this paper suffer from both of these limitations highlighting the need for a more general approach.

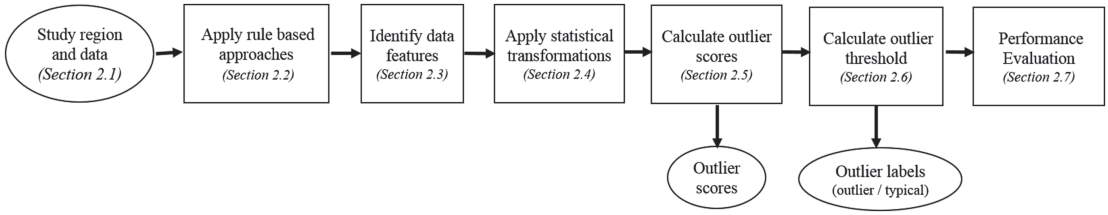


Figure 1. Unsupervised feature-based procedure, named oddwater procedure , for outlier detection in water-quality data from in situ sensors. Squares represents the main steps involved. Circles correspond to input and output.

This paper develops a method for detecting technical outliers in water-quality data derived from in situ sensors. Prior work by Leigh et al. (2019) emphasizes the importance of different anomaly types and end user needs and provides the starting point for constructing a framework for automated anomaly detection in high-frequency water-quality data from in situ sensors. Their work briefly introduced unsupervised feature-based methods for detecting technical outliers in such data. The present paper differs substantially from Leigh et al. (2019) as (1) the unsupervised feature-based procedure we present for detecting technical outliers in high-frequency water-quality data measured by in situ sensors is its sole focus , (2) the unsupervised feature-based procedure is fully elaborated in both details and depth, and (3) the experimental results are enhanced through emphasis on the multivariate capabilities of the unsupervised feature-based procedure. Furthermore, we focus on outliers involving abrupt changes in value, including sudden spikes, sudden isolated drops, and level shifts (high-priority outliers as described in Leigh et al., 2019) rather than the broader suite considered by Leigh et al. (2019).

First, we present in detail our unsupervised feature-based procedure that provides early detection of technical outliers in water-quality data from in situ sensors. Rule-based methods are also incorporated into the procedure to flag occurrences of impossible, out-of-range, and missing values. Second, we provide a comparative analysis of the efficacy and reliability of both density-based and nearest neighbor distance-based outlier scoring techniques. Third, we introduce an R (R Core Team, 2018) package, `oddwater` (Talagala & Hyndman, 2019b), that implements the feature-based procedure and related functions. Further, to facilitate reproducibility and reusability of the results presented in this paper, we have made all of the code and associated data sets available on zenodo (Talagala & Hyndman, 2019a).

Our feature-based procedure has many advantages: (1) It can take the correlation structure of the water-quality variables into account when detecting outliers; (2) it can be applied to both univariate and multivariate problems; (3) the outlier scoring techniques that we consider are unsupervised, data-driven approaches and therefore do not require training data sets for the model building process and can be extended easily to other time series from other sites; (4) the outlier thresholds have a probabilistic interpretation as they are based on extreme value theory; (5) the approach has the ability to deal with irregular (unevenly spaced) time series; and (6) it can easily be extended to streaming data. In contrast to a batch scenario, which assumes that the entire data set is available prior to the analysis with the focus on detecting complete events, the streaming data scenario gives many additional challenges due to high velocity, unbounded, nonstationary data with incomplete events (Hill et al., 2009; Talagala, Hyndman, Smith-Miles, Kandanaarachchi, et al., 2019). In this paper, although our oddwater procedure is introduced as a batch method, it can easily be extended to streaming data such that it can provide near-real-time support using a sliding window technique.

2. Materials and Methods

Our unsupervised feature-based procedure for detecting outliers in water-quality data from in situ sensors has six main steps (Figure 1), and the structure of this section is organized accordingly. For easy reference, we named our unsupervised feature-based procedure as oddwater procedure, which stands for Outlier Detection in Data from WATER-quality sensors.

2.1. Study Region and Data

To evaluate the effectiveness of our oddwater procedure, we considered a challenging real-world problem of monitoring water-quality using in situ sensors in a natural river system. This is challenging because

the system is susceptible to a wide range of environmental, biological, and human impacts that can lead to variation in water quality and affect the technological performance of the sensors. For comparison, we evaluated two study sites, Sandy Creek and Pioneer River, both in the Mackay-Whitsunday region of north-eastern Australia (Mitchell et al., 2005). These two rivers flow into the Great Barrier Reef lagoon and have catchment areas of 1,466 and 326 km^2, respectively. In this region, the wet season typically occurs from December to April and is dominated by higher rainfall and air temperatures, whereas the dry season typically occurs from May to November with lower rainfall and air temperatures (McInnes et al., 2015). The sensors at these two sites are housed within monitoring stations on the river banks. Water is pumped from the rivers to the stations approximately every 60 or 90 min to take measurements of various water-quality variables that are logged by the sensors. Here we focused on three water-quality variables: turbidity (NTU), conductivity (strictly, specific conductance at 25 °C; μS/cm), and river level (m).

The water-quality data obtained from in situ sensors located at Sandy Creek were available from 12 March 2017 to 12 March 2018. The data set included 5,402 recorded points. These time series were irregular (i.e., the frequency of observations was not constant) with a minimum time gap of 10 min and a maximum time gap of around 4 hr. The data obtained from Pioneer River were available from 12 March 2017 to 12 March 2018 and included 6,303 recorded points. Many missing values were observed during the initial part of all three series, that is, turbidity, conductivity, and river level, at Pioneer River. With the help of a group of water-quality experts who were familiar with the study region and with over 40 years of combined knowledge of river water quality, observations were labeled as outliers or not, with the aim of evaluating the performance of the procedure. Our Shiny web application available through the *oddwater* R package was used during the labeling process to pinpoint observations and provide greater visual insight into the data. Using this interactive visualization tool and expert knowledge, the ground-truth labels were decided by consensus vote.

2.2. Apply Rule-Based Approaches

Following Thottan and Ji (2003), we incorporated simple rules into our oddwater procedure to detect outliers such as out-of-range values, impossible values (e.g., negative values), and missing values and labeled them prior to applying the statistical transformations introduced in section 2.4.

If a sensor reading was outside the corresponding sensor detection range, it was marked as an outlier. Negative readings are also inaccurate and impossible for river turbidity, conductivity, and level. We therefore imposed a simple constraint on the algorithm to filter these values and mark them as outliers. Missing values are also frequently encountered in water-quality sensor data (Rangeti et al., 2015). We detected missing values by calculating the time gaps between readings. If a gap exceeded the maximum allowable time difference between any two consecutive readings, the corresponding time stamp was then marked as an outlier due to missingness. Here the maximum allowable time difference was set at 180 min, given that the water-quality measurements were set to be taken at most every 90 min (measurements were often taken at higher frequencies during high-flow events, e.g., every 10–15 min, and occasionally as one-off measurements at times of interest to water managers).

2.3. Identify Data Features

After labeling out-of-range, impossible, and missing values as outliers, further investigation was done with the remaining observations. We initiated this investigation by identifying common characteristics or patterns of the possible types of outliers in water-quality data that would differentiate them from typical instances or events. For turbidity, for example, “extreme” deviations upward are more likely than deviations downward (Panguluri et al., 2009). The opposite is true for conductivity (Tutmez et al., 2006). Further, in a turbidity time series, a sudden isolated upward shift (spike) is a point outlier (a single observation that is surprisingly large, independent of the neighboring observations; Goldstein & Uchida, 2016), but if the sudden upward shift is followed by a gradually decaying tail, then it becomes part of the typical behavior. For river level, rates of rise are often fast compared with fall rates. In general, isolated data points that are outside the general trend are outliers. Further, natural water processes under typical conditions generally tend to be comparatively slow; sudden changes therefore mostly correspond to outlying behaviors. Hereafter, these characteristics will be referred to as “data features.”

2.4. Apply Statistical Transformations

After identifying the data features, different statistical transformations were applied to the time series to highlight different types of outliers focusing on sudden isolated spikes, sudden isolated drops, sudden shifts,

Table 1
Transformation Methods Used to Highlight Different Types of Outliers in Water-Quality Sensor Data

Data feature	Requirement	Possible transformation	Formula
High variability of the data	Stabilize the variance across time series and make the patterns more visible (e.g., level shifts)	Log transformation	$\log(y_t)$
Isolated spikes (in both positive and negative directions) that are outside the general trend are considered as outliers. Under typical behavior, sudden upward (downward) shifts are possible for turbidity (conductivity), but their rate of fall (rise) is generally slower than the rate of rise (fall).	Separate isolated spikes from the general upward/downward trend patterns	First difference	$\log(y_t/y_{t-1})$
Missing values in the data. The maximum allowable time difference between observations is 180 min.	Identify missing values	Time gap	Δt
Data are unevenly spaced time series.	Handle irregular time series	First derivative (Data points with large gaps will get small value. Large gaps indicate the lack of information to make a claim regarding the points.)	$x_t = \log(y_t/y_{t-1})/\Delta t$
Extreme upward trend in turbidity and level under typical behavior.	Separate spikes from typical upward trends.	Turbidity or level	$\min\{x_t, 0\}$
Extreme downward trend in conductivity under typical behavior.	Separate isolated drops from typical downward trends.	Conductivity	$\max\{x_t, 0\}$
High or low variability in the data.	Detect change points in variance.	Rate of change	$(y_t - y_{t-1})/y_t$
Natural processes are comparatively slow. Sudden changes (upward or downward movements) typically correspond to outlying instances.	Detect sudden changes (both upward and downward movements)	Relative difference	$y_t - (1/2)(y_{t-1} + y_{t+1})$

Note. Let Y_t represent an original series from one of the three variables: turbidity, conductivity, and level at time t.

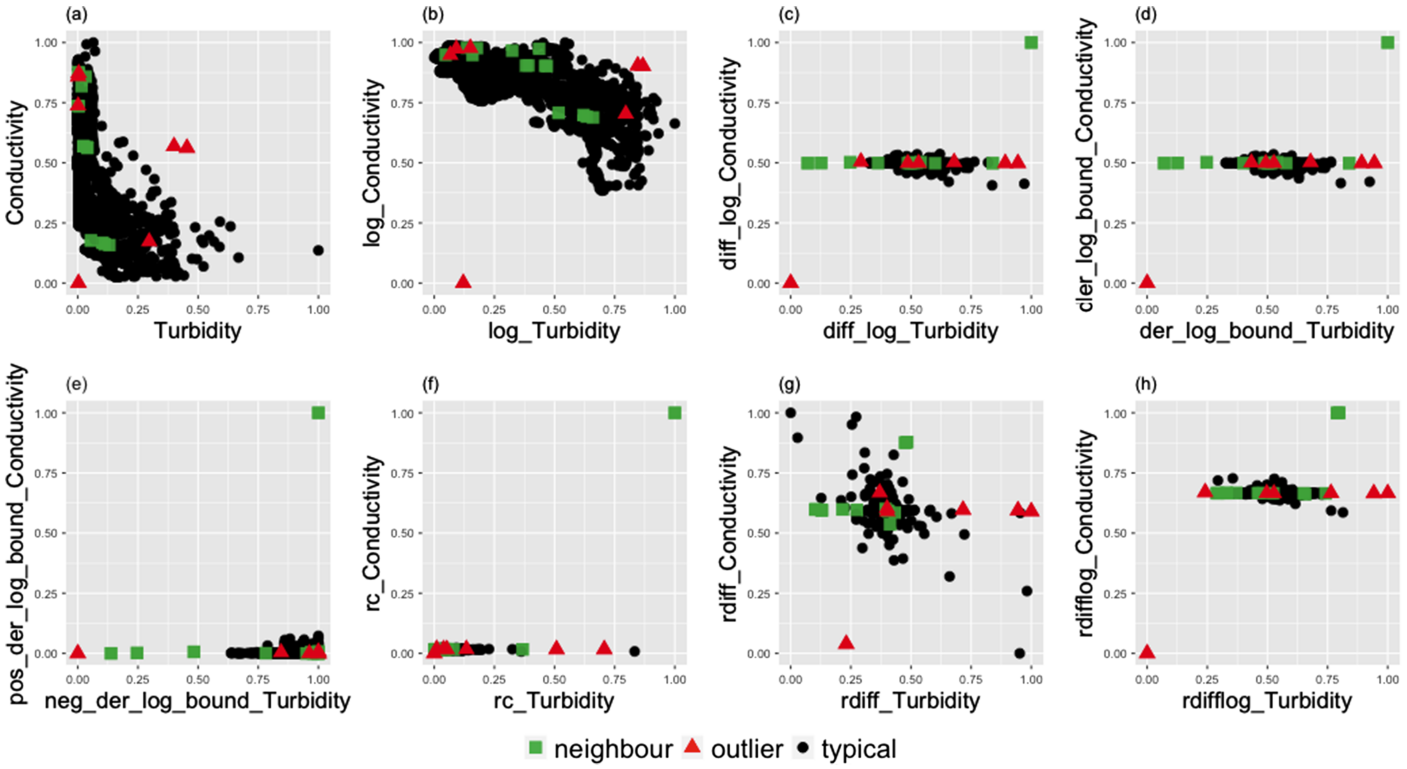


Figure 2. Bivariate relationships between transformed series of turbidity and conductivity measured by in situ sensors at Sandy Creek. In each scatter plot, outliers determined by water-quality experts are shown in red, while typical points are shown in black. Neighboring points are marked in green. (a) Original series, (b) log transformation, (c) first difference, (d) first derivative, (e) one-sided derivative, (f) rate of change, (g) relative difference (for original series), and (h) relative difference (for log-transformed series). In each scatter plot, data are normalized such that they are bounded by the unit hypercube.

and clusters of spikes (Table 1) that deviate from the typical characteristics of each variable (Leigh et al., 2019).

In this work, we considered the outlier detection problem in a multivariate setting. By applying different transformations on water-quality variables, we converted our original problem of outlier detection in the temporal context to a nontemporal context through a high-dimensional data space with three dimensions defined by the three variables: turbidity, conductivity, and river level. Different transformations were applied on different axes of the three-dimensional data space resulting in different data patterns. We evaluated the performance of the transformations (Dang & Wilkinson, 2014) using the maximum separability of the two classes: outliers and typical points in the three-dimensional data space. To provide a better visual illustration, in Figure 2, we present only the two-dimensional data space defined by turbidity and conductivity; however, our actual data space is three dimensional. In this work our focus was to evaluate whether each point in time is an outlier or not such that an alarm could be triggered in the presence of an outlier. However, it was not our interest to investigate which variable(s) is (are) responsible for the outlier in time. Therefore, in Figure 2, a point is marked as an outlier in the two-dimensional space if at least one variable corresponding to that point was labeled as an outlier by the water-quality experts.

When the transformation involves both the current value, Y_t, and the lagged value, Y_{t-1} (as in the first difference and first derivative), both the outlier and immediate neighbor are highlighted in the transformed space. For example, if an outlier occurs at time point t, then the two values derived from the first derivative transformation ($(y_t - y_{t-1})$ and $(y_{t+1} - y_t)$) are highlighted as outlying values, because they both involve y_t. Therefore, each outlying instance is now represented by two consecutive values under the first derivative or first difference transformation. As a result, one outlying instance is now represented by two points in the transformed data space (Figures 2c and 2d). The goal of the one-sided derivative transformation is to select only one high value as a representative point for each outlying instance. However, the high values

obtained could correspond to either the actual outlying time point or the neighboring time point, because each transformed value is derived from two consecutive observations. For example, in the data obtained from Sandy Creek, the one-sided derivative transformation (Figure 2e) clearly separates all of the target outlying instances from the typical points using only one point for each outlying instance, shown as either red triangles (corresponding to outliers) or green squares (corresponding to the immediate neighbors of outliers). The second representative member of each outlying instance mingles with the typical points, allowing only one point to standout on behalf of the corresponding outlying instance. If the primary focus of detecting technical outliers is to alert managers of sensor failures, then it will be inconsequential if the alarm is triggered either at the actual time point corresponding to the outlier or at the next immediate time point. However, if the purpose is different, such as producing a trustworthy data set by labeling or correcting detected outliers, then additional conditions should be imposed to ensure that the time points declared as outliers correspond to the actual outlying points and not to their immediate neighboring points.

2.5. Calculate Outlier Scores

We considered eight commonly used, unsupervised outlier scoring techniques for high-dimensional data involving nearest neighbor distances or densities of the observations and applied them to the three-dimensional data space defined by the three variables: turbidity, conductivity, and river level. Methods based on k-nearest neighbor distances (where $k \in Z^+$) were the NN-HD algorithm (details of this algorithm, which was inspired by HDoutliers algorithm, Wilkinson, 2018, are provided in supporting information S1), KNN-AGG and KNN-SUM algorithms (Angiulli & Pizzuti, 2002; Madsen, 2018), and Local Distance-based Outlier Factor (LDOF) algorithm (Zhang et al., 2009), which calculate the outlier score under the assumption that any outlying point (or outlying clusters of points) in the data space is (are) isolated; therefore, the outliers are those points having the largest k-nearest neighbor distances. In contrast, the density-based Local Outlier Factor (LOF; Breunig et al., 2000), Connectivity-based Outlier Factor (COF; Tang et al., 2002), Influenced Outlierness (INFLO; Jin et al., 2006), and Robust Kernel-based Outlier Factor (Gao et al., 2011) algorithms calculate an outlier score based on how isolated a point is with respect to its surrounding neighbors, and therefore, the outliers are those points having the lowest densities (see supporting information S1 for detail). Each algorithm assigns outlier scores for all of the data points in the high-dimensional space that describe the degree of outlierness of the individual data points such that outliers are those points having the largest scores (Kriegel et al., 2010; Shahid et al., 2015). This step allowed us to set a data-driven threshold (section 2.6) for the outlier scores to select the most relevant outliers (Chandola et al., 2009).

2.6. Calculate Outlier Threshold

Following Schwarz (2008), Burridge and Taylor (2006), and Wilkinson (2018), we used extreme value theory to calculate a separate outlier threshold for each set of outlier scores calculated using a given unsupervised outlier scoring technique (introduced in section 2.5) and assign a bivariate label for each point either as an outlier or typical point. Thus, eight outlier scoring techniques resulted eight different thresholds for a given data set. The threshold calculation process started from a subset of data containing 50% of observations with the smallest outlier scores, under the assumption that this subset contained the outlier scores corresponding to typical data points and the remaining subset contained the scores corresponding to the possible candidates for outliers. Following Weissman's (1978) spacing theorem, the algorithm then fit an exponential distribution to the upper tail of the outlier scores of the first subset and computed the upper $1 - \alpha$ (in this work α was set to 0.05) points of the fitted cumulative distribution function, thereby defining an outlying threshold for the next outlier score. From the remaining subset, the algorithm then selected the point with the smallest outlier score. If this outlier score exceeded the cutoff point, all the points in the remaining subset were flagged as outliers and searching for outliers ceased. Otherwise, the point was declared as a nonoutlier and was added to the subset of the typical points. The threshold was then updated by including the latest addition. The searching algorithm continued until an outlier score was found that exceeded the latest threshold (Schwarz, 2008). We performed this threshold calculation under the assumption that the distribution of outlier scores produced by each of the eight unsupervised outlier scoring techniques for high-dimensional data was in the maximum domain of attraction of the Gumbel distribution, which consists of distribution functions with exponentially decaying tails including the exponential, gamma, normal, and log-normal (Embrechts et al., 2013).

2.7. Performance Evaluation

In this paper, we focused on high-priority outliers as described in Leigh et al. (2019) in which importance ranking of different outlier types was done by taking into account the end user goals and the potential impact

of outliers going undetected. However, it is beyond the scope of this paper to discuss in detail the different types of outliers and their importance ranking. For more detail, we refer the reader to Leigh et al. (2019). We performed an experimental evaluation on the accuracy and computational efficiency of our oddwater procedure with respect to the eight outlier scoring techniques using the different transformations (Table 1) and different combinations of variables (turbidity, conductivity, and river level). These experimental combinations were evaluated with respect to common measures for binary classification based on the values of the confusion matrix, which summarizes the false positives (FP; i.e., when a typical observation is misclassified as an outlier), false negatives (FN; i.e., when an actual outlier is misclassified as a typical observation), true positives (TP; i.e., when an actual outlier is correctly classified), and true negatives (TN; i.e., when an observation is correctly classified as a typical point). In this work, FP and FN are equally undesirable as FP may demand unnecessary and/or expensive actions for corrections and refinement, and FN greatly reduce confidence in the data and results derived from them. The measures we considered include accuracy

$$accuracy = (TP + TN)/(TP + FP + FN + TN), \tag{1}$$

which explains the overall effectiveness of a classifier; and geometric mean

$$GM = \sqrt{TP * TN}, \tag{2}$$

which explains the relative balance of TP and TN of the classifier (Sokolova & Lapalme, 2009). According to Hossin and Sulaiman (2015), these measures are not enough to capture the poor performance of the classifiers in the presence of imbalanced data sets where the size of the typical class (positive class) is much larger than the outlying class (negative class). The data sets obtained from in situ sensors were highly imbalanced and negatively dependent (i.e., containing many more typical observations than outliers). Therefore, we used three additional measures that are recommended for imbalanced problems with only two classes (i.e., typical and outlying) by Ranawana and Palade (2006): the negative predictive value

$$NPV = TN/(FN + TN), \tag{3}$$

which measures the probability of a negatively predicted pattern actually being negative; positive predictive value

$$PPV = TP/(TP + FP), \tag{4}$$

which measures the probability of a positively predicted pattern actually being positive; and optimized precision, which is a combination of accuracy, sensitivity, and specificity metrics (Ranawana & Palade, 2006). The optimized precision is calculated as

$$OP = P - RI, \tag{5}$$

where

$$P = S_p N_n + S_n N_p, \tag{6}$$

$$RI = |S_p - S_n|/(S_p + S_n), \tag{7}$$

$$S_p = TN/(TN + FP), \tag{8}$$

$$S_n = TP/(TP + FN), \tag{9}$$

and N_p and N_n represent the proportion of positives (outliers) and negatives (typical) within the entire data set.

To evaluate the performance of our oddwater procedure, we incorporated additional steps after detecting the outlying time points using the outlying threshold based on extreme value theory. This was done because the time points declared as outliers by the outlying threshold could correspond to either the actual outlying points or to their neighbors. Once the time points were declared as outliers, the corresponding points in the three-dimensional space were further investigated by comparing their positions with respect to the median of the typical points declared by the oddwater procedure. This step allowed us to find the most influential variable for each outlying point. For example, in Figure 2e, the isolated point in the first quadrant is an outlier in the two-dimensional space due to the outlying behavior of the conductivity measurement. This allowed us because the deviation of this point from the median of the typical points (around $(0, 0)$) happens

primarily along the conductivity axis. In contrast, the four isolated points in the third quadrant are outliers due to the outlying behavior of the turbidity measurement because the deviations of the four points from the median of the typical points (around (0, 0)) happen primarily along the turbidity axis. After detecting the most influential variable for each outlying instance in the three-dimensional space, further investigations were carried out separately for each individual outlying instance with respect to the most influential variable detected. This allowed us to see whether the outlying instance was due to a sudden spike or a sudden drop by comparing the direction of the detected points with respect to the mean of its two immediate surrounding neighbors and itself. These additional steps in the oddwater procedure allowed us to trigger an alarm at the actual outlying point in time if the neighboring points were declared as outliers instead of the actual outliers. However, we acknowledge that these additional steps select only the most influential variable, not all of the influential variables in the presence of more than one influential variable. The additional steps were incorporated solely to measure the performance of the oddwater procedure. In practice, because the goal is to trigger an alarm in an occurrence of a technical outlier, it is inconsequential if the alarm is triggered either at the actual time point or at the immediate neighboring time points corresponding to the actual outlier. As such, users of the oddwater procedure can ignore these additional steps.

Using the outlier threshold, our oddwater procedure assigns a bivariate label (either as outlier or typical point) to each observed time point and thereby creates a vector of predicted class labels. That is, if a time point is declared as an outlier by oddwater procedure, then that could be due to at least one variable in the data set. We also declared each time point as an outlier or not based on the labels assigned by the water-quality experts. At a given time point, if at least one variable was labeled as an outlier by the water-quality experts, then the corresponding time point was marked as an outlier, thereby creating a vector of ground-truth labels. Then, the performance measures were calculated based on these two vectors of ground-truth labels and predicted class labels. Thus, this performance evaluation was done with respect to the algorithm's ability to label a point in time as an outlier or not (i.e., a point in time is an outlier if the observed value for any one or more of the three variables measured at that point in time are outliers).

2.8. Software Implementation

The oddwater procedure was implemented in the open source R package `oddwater` (Talagala & Hyndman, 2019b), which provides a growing list of transformation and outlier scoring methods for high-dimensional data together with visualization and performance evaluation techniques. In addition to the implementations available through `oddwater` package, `DDoutlier` package (Madsen, 2018) was also used for outlier score calculations. We measured the computation time (mean execution time) using the `microbenchmark` package (Mersmann, 2018) for different combinations of algorithms, transformations, and variable combinations on 28 core Xeon-E5-2680-v4 @ 2.40GHz servers. We also developed an R Shiny web application (available via `oddwater` R package) to provide interactive visual analytic tools to gain greater insight into the data and perform preliminary investigations of the relationships between water-quality variables at different sites. To facilitate reproducibility of the results presented herein, we have archived a snapshot of version 0.7.0 of the R package on zenodo (Talagala & Hyndman, 2019a) along with the code and data sets used. The latest version and ongoing development of the `oddwater` R package are available from Github (https://github.com/pridiltal/oddwater).

3. Results

3.1. Analysis of Water-Quality Data From In Situ Sensors at Sandy Creek

A negative relationship was clearly visible between the water-quality variables turbidity and conductivity and also between conductivity and river level measured by in situ sensors at Sandy Creek (Figures 3a(i), 3b(i), 3c(i)), 4a, and 4c)). Further, no clear separation was observed between the target outliers and the typical points in the original data space (Figures 4a–4c). However, a clear separation was apparent between the two sets of points once the one-sided derivative transformation (an appropriate transformation for unevenly spaced data) was applied to the original series (Figures 4d–4f, 3a(ii), 3b(ii), and 3c(ii)). KNN-AGG and KNN-SUM algorithms performed on all three water-quality variables together using the one-sided derivative transformation gave the highest OP (0.83) and NPV (0.9996) values, which are the most recommended measurements for negatively dependent data where the focus is more on sensitivity (the proportion of positive patterns being correctly recognized as being positive) than specificity (Ranawana & Palade, 2006).

Based on OP values, the one-sided derivative transformation outperformed the first derivative transformation (Table 2, Rows 1 and 2 compared to Rows 3 and 4). Further, the distance-based outlier detection

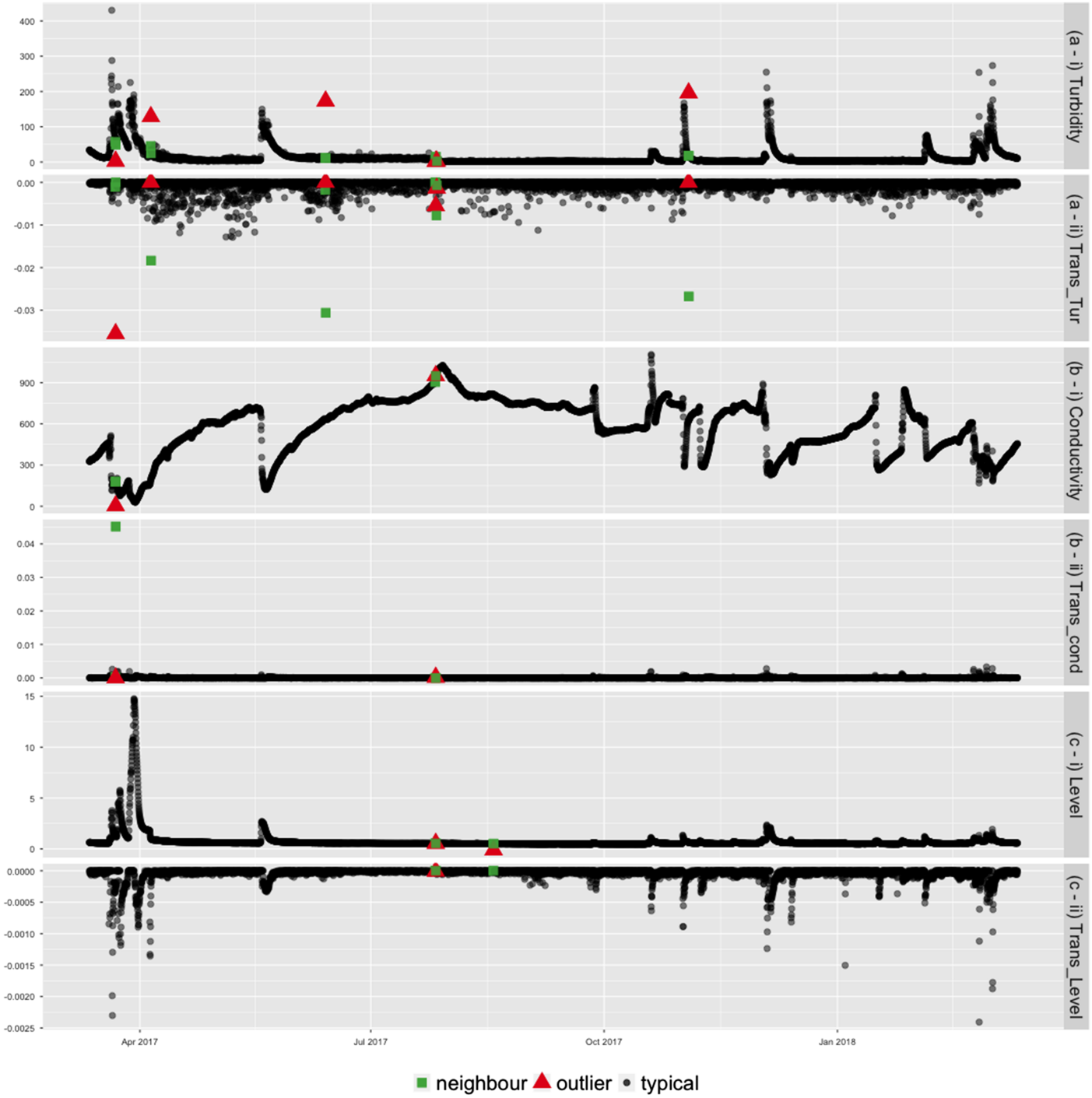


Figure 3. Time series for (a(i)) turbidity (NTU), (b(i)) conductivity (μS/cm), and (c(i)) river level (m) measured by in situ sensors at Sandy Creek. Transformed series (one-sided derivatives) of (a(ii)) turbidity (NTU), (b(ii)) conductivity (μS/cm), and (c(ii)) river level (m) measured by in situ sensors at Sandy Creek. In each plot, outliers determined by water-quality experts are shown in red, while typical points are shown in black. Neighboring points are marked in green.

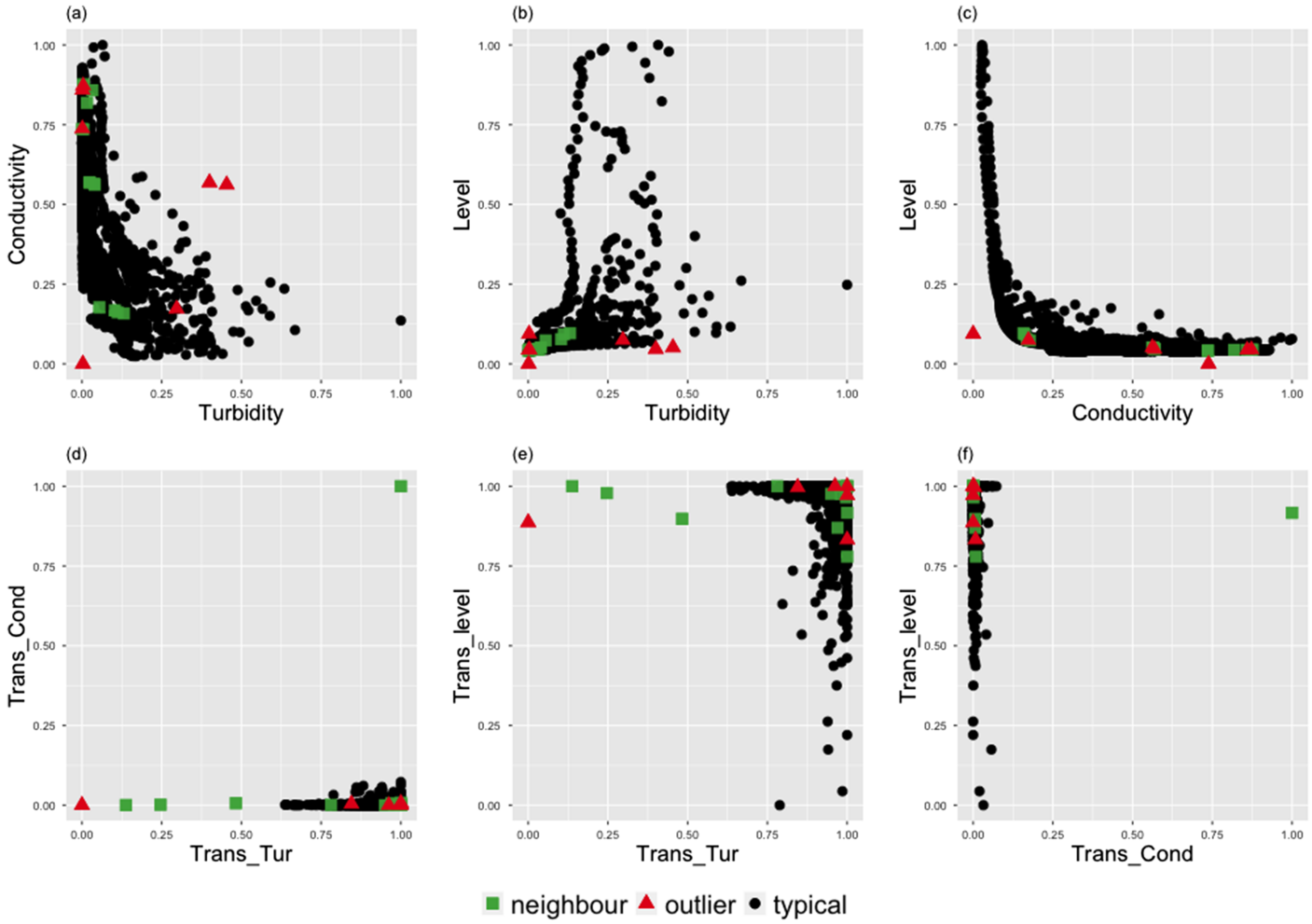


Figure 4. (a–c) Bivariate relationships between original water-quality variables (turbidity [NTU], conductivity [μS/cm], and river level [m]) measured by in situ sensors at Sandy Creek. (d–f) Bivariate relationships between transformed series (one-sided derivative) of turbidity (NTU), conductivity (μS/cm), and river level (m) measured by in situ sensors at Sandy Creek. In each scatter plot, outliers determined by water-quality experts are shown in red, while typical points are shown in black. Neighboring points are marked in green.

algorithms NN-HD, KNN-AGG, and KNN-SUM outperformed all others (Table 2, Rows 1–10 compared to Rows 11–48). Among the three methods, the performance of *k*-nearest neighbor distance-based algorithms were only slightly higher (OP = 0.83) than the NN-HD algorithm (OP = 0.80), which is based only on the nearest neighbor distance. The algorithm combinations with the two highest OP values also had highest NPV (0.9996) and PPV (approximately 0.83). Furthermore, considering river level for the detection of outliers in the water-quality sensors slightly improved the performance (OP = 0.83). Among the analysis with transformed series, LOF with the first derivative transformation performed the least well (OP = 0.25). For most of the outlier detection algorithms (KNN-SUM, KNN-AGG, NN-HD, COF, LOF, and INFLO), the poorest performances were associated with the untransformed original series, having the lowest OP and NPV values, highlighting how data transformation can improve the ability of outlier detection algorithms while maintaining low false detection rates.

The three outlier detection algorithms that demonstrated the highest level of accuracy (NN-HD, KNN-AGG, and KNN-SUM) also outperformed the others with respect to computational time. NN-HD algorithm

Table 2
Performance Metrics of Outlier Detection Algorithms Performed on Multivariate Water-Quality Time Series Data (T = Turbidity; C = Conductivity; L = River Level) From In Situ Sensors at Sandy Creek, Arranged in Descending Order of OP Values

i	Variables	Transformation	Method	Accuracy	GM	OP	PPV	NPV	Time (mean)
1	T-C-L	One-sided derivative	KNN-AGG	0.9994	164.23	0.83	0.83	0.9996	404.0
2	T-C-L	One-sided derivative	KNN-SUM	0.9994	164.23	0.83	0.83	0.9996	186.8
3	T-C	First derivative	NN-HD	0.9991	146.87	0.80	0.57	0.9996	45.0
4	T-C	First derivative	KNN-AGG	0.9989	146.86	0.80	0.50	0.9996	415.8
5	T-C	One-sided derivative	NN-HD	0.9996	146.91	0.80	1.00	0.9996	112.9
6	T-C	One-sided derivative	KNN-AGG	0.9994	146.90	0.80	0.80	0.9996	411.7
7	T-C	One-sided derivative	KNN-SUM	0.9994	146.90	0.80	0.80	0.9996	190.4
8	T-C-L	First derivative	KNN-AGG	0.9993	127.22	0.60	1.00	0.9993	404.4
9	T-C-L	First derivative	KNN-SUM	0.9993	127.22	0.60	1.00	0.9993	188.9
10	T-C	First derivative	KNN-SUM	0.9993	103.88	0.50	1.00	0.9993	189.5
11	T-C	First derivative	LDOF	0.9991	103.87	0.50	0.67	0.9993	17,444.7
12	T-C	One-sided derivative	LDOF	0.9991	103.87	0.50	0.67	0.9993	17,253.8
13	T-C-L	First derivative	NN-HD	0.9991	103.87	0.44	1.00	0.9991	52.5
14	T-C-L	First derivative	INFLO	0.9965	103.74	0.44	0.12	0.9991	1,107.9
15	T-C-L	First derivative	COF	0.9987	103.86	0.44	0.50	0.9991	5,939.8
16	T-C-L	First derivative	RKOF	0.9963	103.73	0.44	0.12	0.9991	369.7
17	T-C-L	One-sided derivative	NN-HD	0.9991	103.87	0.44	1.00	0.9991	118.2
18	T-C-L	One-sided derivative	INFLO	0.9985	103.85	0.44	0.40	0.9991	1,113.6
19	T-C-L	One-sided derivative	COF	0.9987	103.86	0.44	0.50	0.9991	5,787.4
20	T-C-L	One-sided derivative	LDOF	0.9985	103.85	0.44	0.40	0.9991	17,261.9
21	T-C-L	One-sided derivative	LOF	0.9985	103.85	0.44	0.40	0.9991	516.9
22	T-C-L	One-sided derivative	RKOF	0.9976	103.80	0.44	0.20	0.9991	370.5
23	T-C-L	Original series	KNN-AGG	0.9989	103.87	0.44	0.67	0.9991	391.6
24	T-C-L	Original series	INFLO	0.9974	103.79	0.44	0.18	0.9991	1,070.7
25	T-C-L	Original series	LDOF	0.9987	103.86	0.44	0.50	0.9991	17,156.9
26	T-C-L	Original series	RKOF	0.9985	103.85	0.44	0.40	0.9991	354.0
27	T-C	First derivative	INFLO	0.9983	73.43	0.28	0.20	0.9991	1,194.9
28	T-C	First derivative	COF	0.9991	73.46	0.28	1.00	0.9991	5,991.8
29	T-C	First derivative	LOF	0.9987	73.44	0.28	0.33	0.9991	512.3
30	T-C	First derivative	RKOF	0.9983	73.43	0.28	0.20	0.9991	363.2
31	T-C	One-sided derivative	INFLO	0.9987	73.44	0.28	0.33	0.9991	1,207.0
32	T-C	One-sided derivative	COF	0.9987	73.44	0.28	0.33	0.9991	5,880.8
33	T-C	One-sided derivative	LOF	0.9969	73.38	0.28	0.08	0.9991	511.3
34	T-C	One-sided derivative	RKOF	0.9961	73.35	0.28	0.06	0.9991	368.3
35	T-C	Original series	KNN-AGG	0.9989	73.45	0.28	0.50	0.9991	405.1
36	T-C	Original series	INFLO	0.9974	73.40	0.28	0.10	0.9991	1,143.6
37	T-C	Original series	LDOF	0.9987	73.44	0.28	0.33	0.9991	17,022.9
38	T-C	Original series	RKOF	0.9985	73.44	0.28	0.25	0.9991	351.8
39	T-C-L	First derivative	LDOF	0.9989	73.45	0.25	1.00	0.9989	17,323.2
40	T-C-L	First derivative	LOF	0.9989	73.45	0.25	1.00	0.9989	517.1
41	T-C-L	Original series	NN-HD	0.9987	73.44	0.25	0.50	0.9989	48.6
42	T-C-L	Original series	KNN-SUM	0.9989	73.45	0.25	1.00	0.9989	177.3

Table 2 (*continued*)

i	Variables	Transformation	Method	Accuracy	GM	OP	PPV	NPV	Time (mean)
43	T-C-L	Original series	COF	0.9989	73.45	0.25	1.00	0.9989	5,931.7
44	T-C-L	Original series	LOF	0.9989	73.45	0.25	1.00	0.9989	505.0
45	T-C	Original series	NN-HD	0.9987	0.00	0.00	0.00	0.9989	41.7
46	T-C	Original series	KNN-SUM	0.9989	0.00	0.00	NaN	0.9989	184.6
47	T-C	Original series	COF	0.9989	0.00	0.00	NaN	0.9989	5,896.4
48	T-C	Original series	LOF	0.9989	0.00	0.00	NaN	0.9989	502.7

Note. See sections 2.7 and 2.8 for performance metric codes and details.

required the least computational time. Among the remaining two, the mean computational time of KNN-AGG (≈400 ms) was twice that of KNN-SUM's (<200 ms). LOF and its extensions (INFLO, COF, and LDOF) demonstrated the poorest performance with respect computational time (>500 ms on average).

Only KNN-SUM and KNN-AGG assigned high scores to most of the targeted outliers in turbidity, conductivity, and level data transformed using the one-sided derivative (Figures 5a and 5b). For each outlying instance, however, the next immediate neighboring point was assigned the high outlier score instead of the true outlying point. After determining the most influential variable using the additional steps of the algorithm (section 2.7), adjustments were made to correct this to the actual outlier. Because of this correction, the first orange triangle for the True Positive in Figures 5a–5h, for instance, is always plotted next to the high outlier score (corresponding to the neighboring point), pointing to the actual outlier instead of the neighboring point. The outlier scores produced by LOF and COF (Figures 5d and 5e) were unable to capture the outlying behaviors correctly and demonstrated high scattering. In comparison to other outlier scoring algorithms, KNN-SUM algorithm displayed a good compromise between accuracy and computational efficiency (Table 2).

3.2. Analysis of Water-Quality Data From In Situ Sensors at Pioneer River

Compared to Sandy Creek where the river level is mostly less than 1 m with occasional bursts of atypical spikes and flow events resulting in levels up to 14.8 m (Figures 3c–3i), Pioneer River is much deeper with the river level ranging between 13.9 and 16.5 m during the period of study (Figures 6c–6i). Two small dense clusters of points gathered around zero were observed for all three variables from late March to mid-April in 2017 (Figure 6). These co-occurrences of values around zero are atypical behavior and may have been due to technical issues with the sensor equipment. These type of anomalies can be easily detected by incorporating rule-based methods.

Some of the target outliers in the data obtained from the in situ sensors at Pioneer River only deviated slightly from the general trend (Figures 6a–6i), making outlier detection challenging. A negative relationship was clearly visible between turbidity and conductivity (Figure 7a); however, the relationship between level and conductivity was complex (Figure 7c). Most of the target outliers were masked by the typical points in the original space (Figures 7a–7c). Similar to Sandy Creek, data obtained from the sensors at Pioneer River showed good separation between outliers and typical points under the one-sided derivative transformation (Figures 7d–7f, 6a(ii), 6b(ii), and 6c(ii)). However, the sudden spikes in turbidity labeled as outliers by water-quality experts could not be separated from the majority by a large distance and were only visible as a small group (microcluster; Goldstein & Uchida, 2016) in the boundary defined by the typical points (Figures 7d and 7e).

From the performance analysis, it was observed that turbidity and conductivity together produced better results (Table 3, Rows 1–8) than when combined with river level, which tended to reduce the performance (i.e., generating lower OP and NPV values) while increasing the false negative rate (Table 3, Rows 9–13). KNN-AGG and KNN-SUM (Table 3, Rows 2 and 3) had the highest accuracy (0.9978), highest geometric means (492.8012), highest OP (0.88), and highest NPV (0.9984). Despite the challenge given by the small spikes which could not be clearly separated from the typical points, KNN-AGG, KNN-SUM, and NN-HD with one-sided derivatives of turbidity and conductivity still detected some of those points as

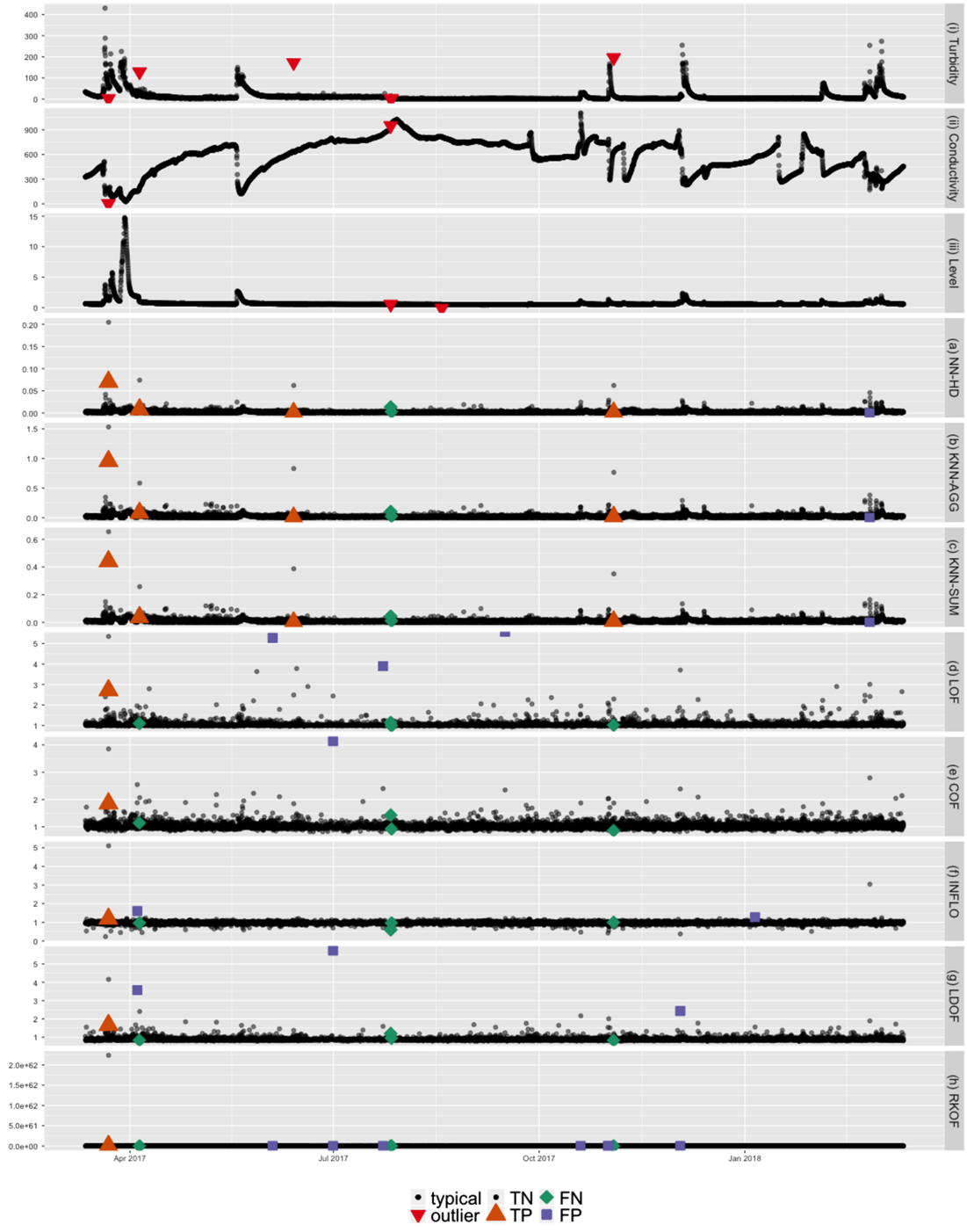


Figure 5. Classification of outlier scores produced from different algorithms as true negatives (TN), true positives (TP), false negatives (FN), and false positives (FP). The top three panels (i–iii) correspond to the original series (turbidity, conductivity, and river level) measured by in situ sensors at Sandy Creek. The target outliers (detected by water-quality experts) are shown in red, while typical points are shown in black. (a)–(h) give outlier scores produced by different outlier detection algorithms for high-dimensional data when applied to the transformed series (one-sided derivative) of the three variables: turbidity, conductivity, and level. Through different outlier scoring algorithms (a–h), we are evaluating whether each point in time is an outlier or not. Therefore, (a)–(h), if the outlier scoring algorithm is effective, then there should be either TP or TN at each point in time when either a red triangle is plotted in at least one of the three panels (i–iii) or black dots are plotted in all of the top three panels (i–iii). Because outlier scores are nonnegative and are mostly clustered near zero, with some occasional high values, a square root transformation was applied to reduce skewness of the data in (a) to (h).

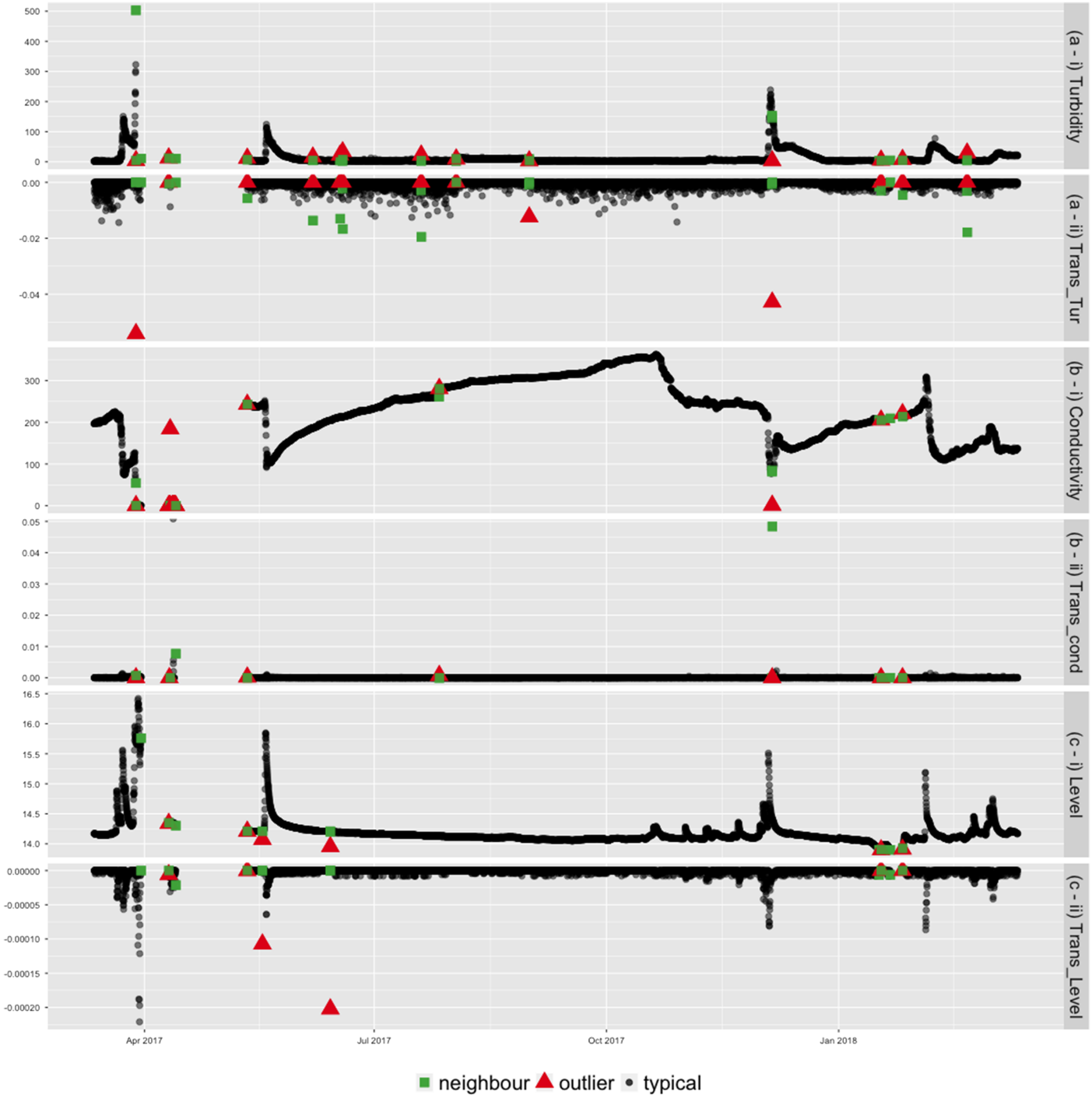


Figure 6. Time series for (a(i)) turbidity (NTU), (b(i)) conductivity (μS/cm), and (c(i)) river level (m) measured by in situ sensors at Pioneer River. Transformed series (one-sided derivatives) of (a(ii)) turbidity (NTU), (b(ii)) conductivity (μS/cm), and (c(ii)) river level (m) measured by in situ sensors at Pioneer River. In each plot, outliers determined by water-quality experts are shown in red, while typical points are shown in black. Neighboring points are marked in green.

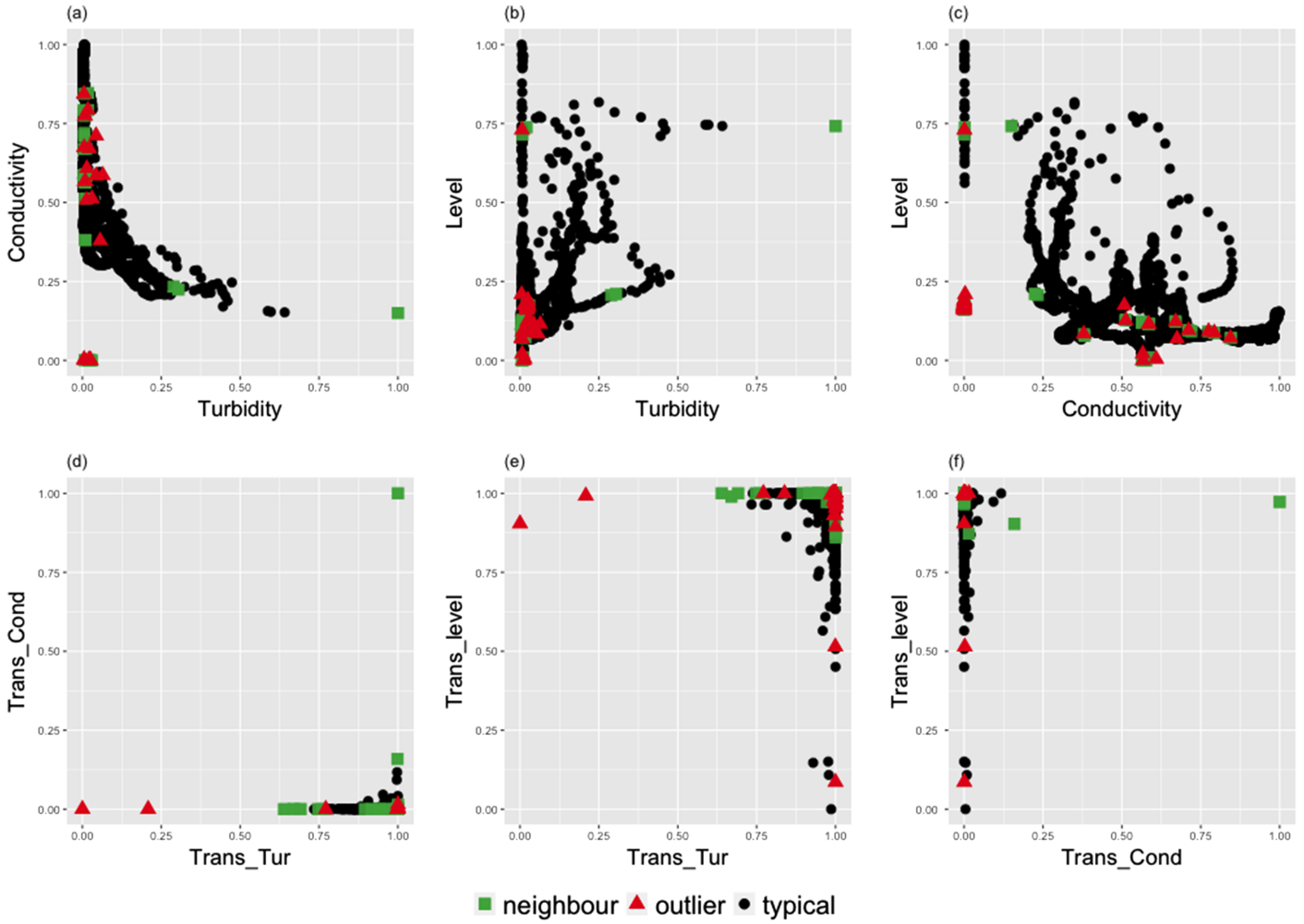


Figure 7. (a–c) Bivariate relationships between original water-quality variables (turbidity [NTU], conductivity [μS/cm], and river level [m]) measured by in situ sensors at Pioneer River. (d–f) Bivariate relationships between transformed series (one-sided derivative) of turbidity (NTU), conductivity (μS/cm), and river level (m) measured by in situ sensors at Pioneer River. In each scatter plot, outliers determined by water-quality experts are shown in red, while typical points are shown in black. Neighboring points are marked in green.

outliers while maintaining low false negative and false positive rates (Figure 8). Similar to Sandy Creek, NN-HD (<200 ms on average) and KNN-SUM (<230 milliseconds on average) demonstrated the highest computational efficiency for the data obtained from Pioneer River.

4. Discussion

We introduced a new procedure, named oddwater procedure, for the detection of outliers in water-quality data from in situ sensors, where outliers were specifically defined as due to technical errors that make the data unreliable and untrustworthy. We showed that our oddwater procedure, with carefully selected data transformation methods derived from data features, can greatly assist in increasing the performance of a range of existing outlier detection algorithms. Our oddwater procedure and analysis using data obtained from in situ sensors positioned at two study sites, Sandy Creek and Pioneer River, performed well with outlier types such as sudden isolated spikes, sudden isolated drops, and level shifts while maintaining low false detection rates. As an unsupervised procedure, our approach can be easily extended to other water-quality variables, other sites, and also to other outlier detection tasks in other application domains. The only requirement is to select suitable transformation methods according to the data features that differentiate the outlying instances from the typical behaviors of a given system.

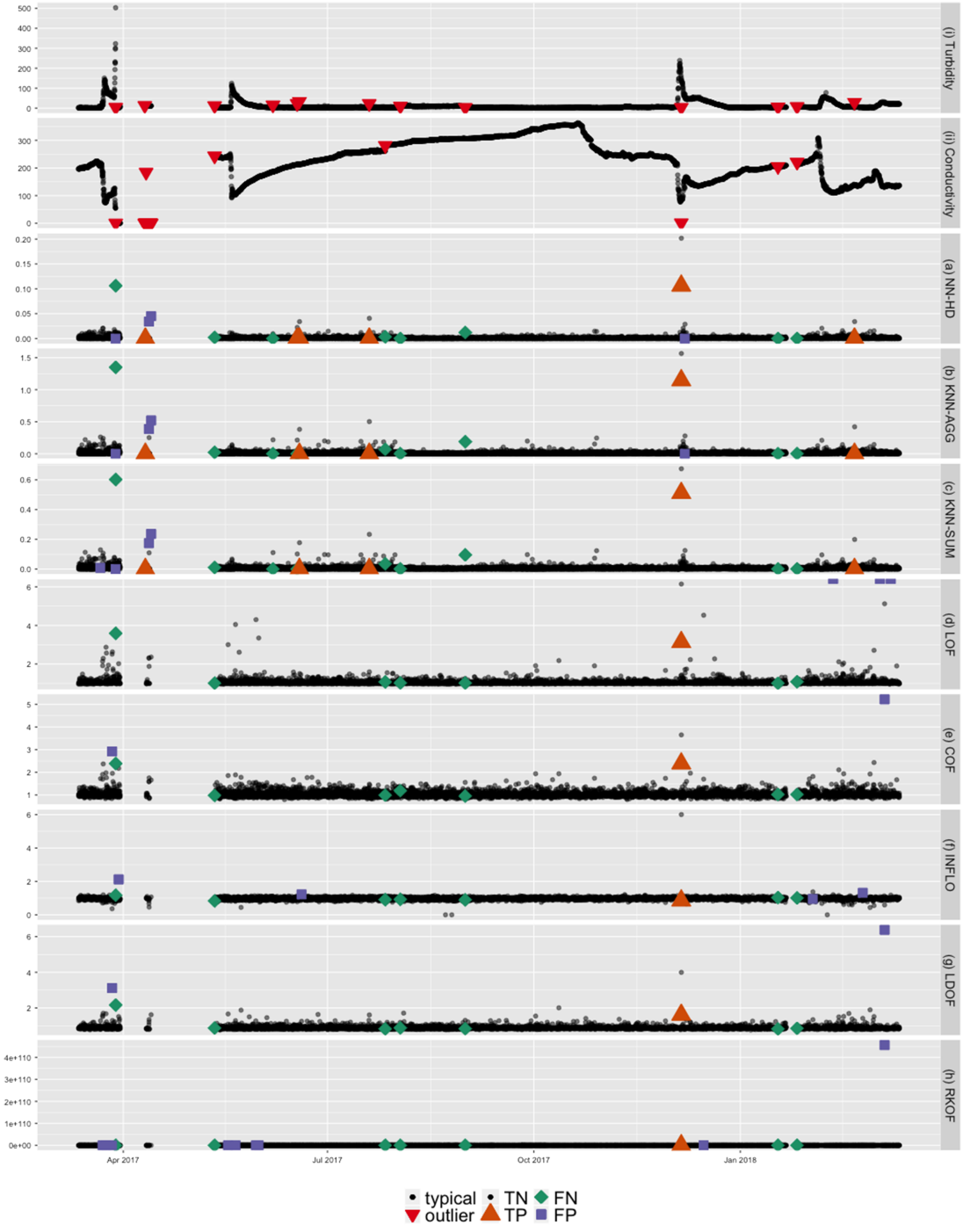


Figure 8. Classification of outlier scores produced from different algorithms as true negatives (TN), true positives (TP), false negatives (FN), and false positives (FP). The top two panels (i and ii) correspond to the original series (turbidity and conductivity) measured by in situ sensors at Pioneer River. The target outliers (detected by water-quality experts) are shown in red, while typical points are shown in black. (a)–(h) give outlier scores produced by different outlier detection algorithms for high-dimensional data when applied to the transformed series (one-sided derivative) of the two variables: turbidity and conductivity. Through different outlier scoring algorithms (a–h), we are evaluating whether each point in time is an outlier or not. Therefore, from (a)–(h), if the outlier scoring algorithm is effective, then there should be either TP or TN at each point in time when either a red triangle is plotted in at least one of the two panels (i and ii) or black dots are plotted in both of the top two panels (i and ii). Because outlier scores are nonnegative and are mostly clustered near zero, with some occasional high values, a square root transformation was applied to reduce skewness of the data in (a) to (h).

Table 3
Performance Metrics of Outlier Detection Algorithms Performed on Multivariate Water-Quality Time Series Data (T = Turbidity; C = Conductivity; L = River Level) From In Situ Sensors at Pioneer River, Arranged in Descending Order of OP Values

i	Variables	Transformation	Method	Accuracy	GM	OP	PPV	NPV	Time (mean)
1	T-C	One-sided derivative	NN-HD	0.9976	492.76	0.88	0.89	0.9984	136.5
2	T-C	One-sided derivative	KNN-AGG	0.9978	492.80	0.88	0.91	0.9984	478.8
3	T-C	One-sided derivative	KNN-SUM	0.9978	492.80	0.88	0.91	0.9984	222.2
4	T-C	First derivative	NN-HD	0.9978	480.08	0.86	0.95	0.9981	182.0
5	T-C	First derivative	KNN-AGG	0.9978	480.08	0.86	0.95	0.9981	488.5
6	T-C	First derivative	KNN-SUM	0.9978	480.08	0.86	0.95	0.9981	225.3
7	T-C	First derivative	INFLO	0.9971	479.92	0.86	0.86	0.9981	1,525.0
8	T-C	First derivative	RKOF	0.9970	479.88	0.86	0.84	0.9981	430.4
9	T-C-L	One-sided derivative	KNN-AGG	0.9975	492.72	0.86	0.91	0.9981	465.2
10	T-C-L	One-sided derivative	KNN-SUM	0.9975	492.72	0.86	0.91	0.9981	214.5
11	T-C-L	First derivative	RKOF	0.9951	485.82	0.85	0.68	0.9979	425.9
12	T-C-L	First derivative	KNN-AGG	0.9975	480.00	0.84	0.95	0.9978	478.0
13	T-C-L	First derivative	KNN-SUM	0.9975	480.00	0.84	0.95	0.9978	220.0
14	T-C	First derivative	COF	0.9978	473.58	0.84	0.97	0.9979	7,908.2
15	T-C	First derivative	LDOF	0.9978	473.58	0.84	0.97	0.9979	23,435.7
16	T-C	First derivative	LOF	0.9975	473.51	0.84	0.92	0.9979	594.4
17	T-C	One-sided derivative	INFLO	0.9973	473.47	0.84	0.90	0.9979	1,559.9
18	T-C	One-sided derivative	COF	0.9976	473.54	0.84	0.95	0.9979	7,505.5
19	T-C	One-sided derivative	LDOF	0.9975	473.51	0.84	0.92	0.9979	22,986.0
20	T-C	One-sided derivative	LOF	0.9975	473.51	0.84	0.92	0.9979	596.9
21	T-C	One-sided derivative	RKOF	0.9960	473.16	0.84	0.75	0.9979	419.7
22	T-C	Original series	INFLO	0.9973	473.47	0.84	0.90	0.9979	1,498.5
23	T-C-L	First derivative	COF	0.9975	473.51	0.83	0.97	0.9976	7,910.7
24	T-C-L	First derivative	LDOF	0.9975	473.51	0.83	0.97	0.9976	23,357.7
25	T-C-L	One-sided derivative	NN-HD	0.9975	473.51	0.83	0.97	0.9976	131.9
26	T-C	Original series	NN-HD	0.9976	466.96	0.83	0.97	0.9978	171.0
27	T-C	Original series	KNN-AGG	0.9970	466.81	0.83	0.88	0.9978	468.7
28	T-C	Original series	KNN-SUM	0.9970	466.81	0.83	0.88	0.9978	211.6
29	T-C	Original series	COF	0.9978	467.00	0.83	1.00	0.9978	7,617.6
30	T-C	Original series	LDOF	0.9978	467.00	0.83	1.00	0.9978	22,910.4
31	T-C	Original series	LOF	0.9978	467.00	0.83	1.00	0.9978	579.1
32	T-C	Original series	RKOF	0.9963	466.66	0.83	0.80	0.9978	401.9
33	T-C-L	First derivative	NN-HD	0.9973	473.47	0.82	0.95	0.9976	167.1
34	T-C-L	One-sided derivative	INFLO	0.9971	473.43	0.82	0.92	0.9976	1,418.8
35	T-C-L	One-sided derivative	COF	0.9973	473.47	0.82	0.95	0.9976	7,497.9
36	T-C-L	One-sided derivative	LDOF	0.9973	473.47	0.82	0.95	0.9976	23,090.7
37	T-C-L	One-sided derivative	RKOF	0.9952	472.97	0.82	0.71	0.9976	422.1
38	T-C-L	First derivative	INFLO	0.9975	466.92	0.81	1.00	0.9974	1,398.3
39	T-C-L	First derivative	LOF	0.9975	466.92	0.81	1.00	0.9974	600.7
40	T-C-L	One-sided derivative	LOF	0.9965	466.70	0.81	0.85	0.9974	596.1
41	T-C-L	Original series	NN-HD	0.9973	466.88	0.81	0.97	0.9974	163.0
42	T-C-L	Original series	KNN-AGG	0.9967	466.73	0.81	0.88	0.9974	456.3

Table 3 (*continued*)

i	Variables	Transformation	Method	Accuracy	GM	OP	PPV	NPV	Time (mean)
43	T-C-L	Original series	KNN-SUM	0.9967	466.73	0.81	0.88	0.9974	201.4
44	T-C-L	Original series	INFLO	0.9975	466.92	0.81	1.00	0.9974	1,372.8
45	T-C-L	Original series	COF	0.9975	466.92	0.81	1.00	0.9974	7,707.2
46	T-C-L	Original series	LDOF	0.9975	466.92	0.81	1.00	0.9974	127,337.1
47	T-C-L	Original series	LOF	0.9975	466.92	0.81	1.00	0.9974	580.9
48	T-C-L	Original series	RKOF	0.9955	466.47	0.81	0.74	0.9974	406.8

Note. See sections 2.7 and 2.8 for performance metric codes and details.

Studies have shown that transforming variables affects densities, relative distances, and orientation of points within the data space and therefore can improve the ability to perceive patterns in the data which are not clearly visible in the original data space (Dang & Wilkinson, 2014). This was the case in our study where no clear separation was visible between outliers and typical data points in the original data space, but a clear separation was obtained between the two sets of points once the one-sided derivative transformation was applied to the original series. Having this type of a separation between outliers and typical points is important before applying unsupervised outlier detection algorithms for high-dimensional data because the methods are usually based on the definition of outliers in terms of distance or density (Talagala, Hyndman, Smith-Miles, Kandanaarachchi, et al., 2019). Most of the outlier detection algorithms (KNN-SUM, KNN-AGG, NN-HD, COF, LOF, and INFLO) performed least well with the untransformed original series, demonstrating how data transformation methods can assist in improving the ability of outlier detection algorithms while maintaining low false detection rates.

In our modified algorithm, the NN-HD algorithm, we did not incorporate the clustering step of the HDoutliers algorithm because the data obtained from the two study sites are free from microclusters (Talagala, Hyndman, Smith-Miles, et al., 2019) and therefore free from the masking problem. Because the data sets have only local and global outliers, incorporating a clustering step that forms small clusters using a small ball with a fixed radius (the Leader Algorithm in Wilkinson, 2018) does not significantly change the structure of the data points in the high-dimensional data space. Furthermore, because NN-HD has the additional requirement of isolation in addition to clear separation between outlying points and typical points, it performed poorly in comparison to the two KNN distance-based algorithms (KNN-AGG and KNN-SUM) which are not restricted to the single most nearest neighbor (Talagala, Hyndman, Smith-Miles, et al., 2019). For the current work, k was set to 10, the maximum default value of k in Madsen (2018), because too large a value of k could skew the focus toward global outliers (points that deviates significantly from the rest of the data set) alone (Zhang et al., 2009) and make the algorithms computationally inefficient. On the other hand, too small a value of k could incorporate an additional assumption of isolation into the algorithm, as in the NN-HD algorithm where $k = 1$. Among the analyses using transformed series, LOF with the first derivative transformation performed the least well, which could also be due to its additional assumption of isolation (Tang et al., 2002). However, using the same k across all algorithms may bias direct comparison because the performance of the algorithms can depend on the value of k and algorithms can reach their peak performance for different choices of k (Campos et al., 2016). Therefore, performing an optimization to select the best k is nontrivial, and we leave it for future work.

We took the correlation structure between the variables into account when detecting outliers given some were apparent only in the high-dimensional space but not when each variable was considered independently (Ben-Gal, 2005). A negative relationship was observed between conductivity and turbidity and also between conductivity and level for the Sandy Creek data. However, for Pioneer River, no clear relationship was observed between level and the remaining two variables, turbidity, and conductivity. This could be one reason why the variable combination with river level gave poor results for the Pioneer River data set, while results for other combinations were similar to those of Sandy Creek. The one-sided derivative transformation outperformed the derivative transformation. This was expected, because in an occurrence of a sudden spike or isolated drop, the first derivative assigns high values to two consecutive points, the actual outlying point and the neighboring point, and therefore increases the false positive rate (because the neighboring points that are declared to be outliers actually correspond to typical points in the original data space). Therefore, to detect technical outliers in water-quality data from Sandy Creek and Pioneer River, the one-sided derivative

transformation is recommended because it outperformed the other transformations during the comparative analysis. For Sandy Creek, all three water-quality variables together with the one-sided derivative transformation is recommended. However, for Pioneer River, the use of river level is not advisable due its complex relationships with the other variables and its temporal variability. For both rivers, the use of KNN-SUM algorithm is recommended because it provides a good compromise between accuracy and computational efficiency.

In this study, our goal was to detect suitable transformations, combinations of variables, and the algorithms for outlier score calculation for the data from two study sites. Results may depend on the characteristics of the time series (site and time dependent for example), and what is best for one site may not be the best for another site. Therefore, care should be taken to select transformations most suitable for the problem at hand. According to Dang and Wilkinson (2014), any transformation used on a data set must be evaluated in terms of a figure of merit (i.e., a numerical quantity used to characterize the performance of a method, relative to its alternatives). For our work on detecting outliers, the figure of merit was the maximum separability of the two classes generated by outliers and typical points. However, we acknowledge that the set of transformations that we used for this work was relatively limited and influenced by the data obtained from the two study sites. Therefore, the set of transformations we considered (Table 1) should be viewed only as an illustration of our oddwater procedure for detecting outliers. We expect that the set of transformations will expand over time as the oddwater procedure is used for other data from other study sites and for applications to other fields.

For the current work, we selected transformation methods that could highlight abrupt changes in the water-quality data. We hope to expand the ability of oddwater procedure so that it can detect other outlier types not previously targeted but commonly observed in water-quality data (e.g., low/high variability and drift as per Leigh et al., 2019). One possibility is to consider the residuals at each point, defined as the difference between the actual values and the fitted values (similar to Schwarz, 2008) or the difference between the actual values and the predicted values (similar to Hill & Minsker, 2006), as a transformation and apply outlier detection algorithms to the high-dimensional space defined by those residuals. Here the challenge will be to identify the appropriate curve fitting and prediction models to generate the residual series. In this way, continuous subsequences of high values could correspond to other kinds of technical outliers such as high variability or drift. However, the range of applications and the space of the transformations are extremely diverse, which makes it challenging to provide a structured formal vision that covers all of the possible transformations that could be considered. The transformations we present in this paper were mainly chosen as appropriate to the data collected from Sandy Creek and Pioneer River. We observed that different transformations can lead to entirely different data structures and that the selection of suitable transformations is directed by the data features and typical patterns imposed by a given application. Domain specific knowledge plays a vital role when selecting suitable transformations and, as such, defining structured guidelines for the selection of suitable transformations remains problematic.

Not surprisingly, NN-HD algorithm required the least computational time given the outlying score calculation only involves searching for the single most nearest neighbors of each test point (Wilkinson, 2018). The mean computational time of KNN-AGG was twice as high as that of KNN-SUM because the KNN-AGG algorithm has the additional requirement of calculating weights that assign nearest neighbors higher weight relative to the neighbors farther apart (Angiulli & Pizzuti, 2002). LOF and its extensions (INFLO, COF, and LDOF) required the most computational time; all four algorithms involve a two-step searching mechanism at each test point when calculating the corresponding outlying score. This means that at each test point, each algorithm searches its k nearest neighbors as well those of the detected nearest neighbors for the outlier score calculation (Breunig et al., 2000; Jin et al., 2006; Tang et al., 2002; Zhang et al., 2009).

Assessing performance of the detection methods based on the classification criteria, while traditional, has limitations . During performance evaluation, we observed that some outliers were detected by all the approaches, some were detected as outliers only by certain methods, and some were identified by no method. Therefore, incorporating ensemble methods as proposed in Unwin (2019) would assist in selecting the best performing approaches for a particular outlier type and enable further insight into the results obtained from the oddwater procedure.

We hope to extend our multivariate outlier detection framework into space and time so that it can deal with the spatiotemporal correlation structure along branching river networks. Further, in the current paper, we

have introduced our oddwater procedure as a batch method. However, due to the unsupervised nature of our oddwater procedure, it can be easily extended to a streaming data scenario with the help of a sliding window of fixed length. A streaming data scenario always demands a near-real-time support. Therefore, one significant challenge is to find efficient methods that allow us to update outlier scores taking account of the newest observations and removing the oldest observations introduced by overlapping sliding windows, rather than recalculating scores corresponding to observations which are not affected by either new arrivals or the oldest observations (that are no longer covered by the latest window). Further work will be needed to investigate the efficient computation of regenerating nearest neighbors in a data streaming context.

Notation

FP False positives (i.e., when a typical observation is misclassified as an outlier)
FN False negatives (i.e., when an actual outlier is misclassified as a typical observation)
TP True positives (i.e., when an actual outlier is correctly classified)
TN True negatives (i.e., when an observation is correctly classified as a typical point)

Acknowledgments
Funding for this project was provided by the Queensland Department of Environment and Science (DES) and the ARC Centre of Excellence for Mathematical and Statistical Frontiers (ACEMS). The authors would like to acknowledge the Queensland Department of Environment and Science, in particular, the Great Barrier Reef Catchment Loads Monitoring Program for the data and the staff from Water Quality and Investigations for their input. We thank Ryan S. Turner and Erin E. Peterson for several valuable discussions regarding project requirements and water-quality characteristics. Further, 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. We would also like to thank David Hill and other anonymous reviewers for their valuable comments and suggestions. The data sets used for this article are available in the open source R package `oddwater` (Talagala & Hyndman, 2019b).

References

Angiulli, F., & Pizzuti, C. (2002). Fast outlier detection in high dimensional spaces, *European conference on principles of data mining and knowledge discovery* (pp. 15–27). Berlin, Heidelberg: Springer.
Archer, C., Baptista, A., & Leen, T. K. (2003). Fault detection for salinity sensors in the Columbia estuary. *Water Resources Research*, *39*(3), 1060. https://doi.org/10.1029/2002WR001376
Ben-Gal, I. (2005). Outlier detection. In O. Maimon & L. Rokach (Eds.), *Data mining and knowledge discovery handbook* (pp. 131–146). Boston, MA: Springer.
Breunig, M. M., Kriegel, H.-P., Ng, R. T., & Sander, J. (2000). LOF: Identifying density-based local outliers, *ACM Sigmod record* (Vol. 29, pp. 93–104). New York, NY, USA: ACM.
Burridge, P., & Taylor, A. M. R. (2006). Additive outlier detection via extreme-value theory. *Journal of Time Series Analysis*, *27*(5), 685–701.
Campos, G. O., Zimek, A., Sander, J., Campello, R. J., Micenková, B., Schubert, E., et al. (2016). On the evaluation of unsupervised outlier detection: Measures, datasets, and an empirical study. *Data Mining and Knowledge Discovery*, *30*(4), 891–927.
Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM computing surveys (CSUR)*, *41*(3), 15.
Dang, T. N., & Wilkinson, L. (2014). Transforming scagnostics to reveal hidden features. *IEEE transactions on visualization and computer graphics*, *20*(12), 1624–1632.
Embrechts, P., Klüppelberg, C., & Mikosch, T. (2013). *Modelling extremal events: For insurance and finance*, Stochastic Modelling and Applied Probability. Berlin, Heidelberg: Springer. Retrieved from https://books.google.com.au/books?id=BXOI2pICfJUC
Gao, J., Hu, W., Zhang, Z. M., Zhang, X., & Wu, O. (2011). RKOF: Robust kernel-based local outlier detection. In *Pacific-Asia conference on knowledge discovery and data mining* (pp. 270–283). Berlin, Heidelberg: Springer.
Glasgow, H. B., Burkholder, J. M., Reed, R. E., Lewitus, A. J., & Kleinman, J. E. (2004). Real-time remote monitoring of water quality: A review of current applications, and advancements in sensor, telemetry, and computing technologies. *Journal of Experimental Marine Biology and Ecology*, *300*(1-2), 409–448.
Goldstein, M., & Uchida, S. (2016). A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PloS One*, *11*(4), e0152173.
Hill, D. J., & Minsker, B. S. (2006). Automated fault detection for in-situ environmental sensors. In P. Gourbesville, J. Cunge, & S.-Y. Liong (Eds.), *Hydroinformatics 2006: Proceedings of the 7th International Conference on Hydroinformatics*. Chennai, India: Res. Publ. Serv.
Hill, D. J., Minsker, B. S., & Amir, E. (2009). Real-time Bayesian anomaly detection in streaming environmental data. *Water Resources Research*, *45*, W00D28. https://doi.org/10.1029/2008WR006956
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.
Jin, W., Tung, A. K., Han, J., & Wang, W. (2006). Ranking outliers using symmetric neighborhood relationship. In *Pacific-Asia conference on knowledge discovery and data mining* (pp. 577–593). Berlin, Heidelberg: Springer.
Koch, M. W., & McKenna, S. A. (2010). Distributed sensor fusion in water quality event detection. *Journal of Water Resources Planning and Management*, *137*(1), 10–19.
Kotamäki, N., Thessler, S., Koskiaho, J., Hannukkala, A. O., Huitu, H., Huttula, T., et al. (2009). Wireless in-situ sensor network for agriculture and water monitoring on a river basin scale in Southern Finland: Evaluation from a data user perspective. *Sensors*, *9*(4), 2862–2883.
Kriegel, H.-P., Kröger, P., & Zimek, A. (2010). Outlier detection techniques. *Tutorial at KDD*, *10*, 1–73.
Leigh, C., Alsibai, O., Hyndman, R. J., Kandanaarachchi, S., King, O. C., McGree, J. M., et al. (2019). A framework for automated anomaly detection in high frequency water-quality data from in situ sensors. *Science of The Total Environment*, *664*, 885–898.
Madsen, J. H. (2018). Ddoutlier: Distance and density-based outlier detection. Retrieved from https://CRAN.R-project.org/package=DDoutlier (R package version 0.1.0).
McInnes, K., Abbs, D., Bhend, J., Chiew, F., Church, J., Ekstrm, M., et al. (2015). Wet tropics cluster report. In M. Ekström et al. (Eds.), *Climate change in Australia Projections for Australia's Natural Resource Management Regions: Cluster Reports*. Australia: CSIRO and Bureau of Meteorology.
McKenna, S. A., Hart, D., Klise, K., Cruz, V., & Wilson, M. (2007). Event detection from water quality time series. In *World environmental and water resources congress 2007: Restoring our natural habitat* (pp. 1–12). Tampa, Florida, United States: American Society of Civil Engineers.
Mersmann, O. (2018). microbenchmark: Accurate timing functions. https://CRAN.R-project.org/package=microbenchmark (R package version 1.4-4).

Mitchell, C., Brodie, J., & White, I. (2005). Sediments, nutrients and pesticide residues in event flow conditions in streams of the Mackay Whitsunday Region, Australia. *Marine Pollution Bulletin*, *51*(1-4), 23–36.
Moatar, F., Fessant, F., & Poirel, A. (1999). pH modelling by neural networks. Application of control and validation data series in the Middle Loire river. *Ecological Modelling*, *120*(2-3), 141–156.
Moatar, F., Miquel, J., & Poirel, A. (2001). A quality-control method for physical and chemical monitoring data. Application to dissolved oxygen levels in the River Loire (France). *Journal of Hydrology*, *252*(1-4), 25–36.
Panguluri, S., Meiners, G., Hall, J., & Szabo, J. (2009). Distribution system water quality monitoring: Sensor technology evaluation methodology and results (Tech. Rep. EPA/600/R-09/076, 2772). Washington, DC, USA: US Environ. Protection Agency.
R Core Team (2018). R: A language and environment for statistical computing (Computer software manual). Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/
Raciti, M., Cucurull, J., & Nadjm-Tehrani, S. (2012). Anomaly detection in water management systems, *Critical infrastructure protection* (pp. 98–119). Berlin, Heidelberg: Springer.
Ranawana, R., & Palade, V. (2006). Optimized precision—A new measure for classifier performance evaluation. In *IEEE congress on evolutionary computation, 2006. cec 2006*, IEEE, Vancouver, BC, Canada, pp. 2254–2261.
Rangeti, I., Dzwairo, B., Barratt, G. J., & Otieno, F. A. (2015). Validity and errors in water quality data-a review. In *Research and practices in water quality* (pp. 95–112). Durban, South Africa: Durban University of Technology.
Schwarz, K. T. (2008). *Wind dispersion of carbon dioxide leaking from underground sequestration, and outlier detection in eddy covariance data using extreme value theory*. Berkeley: University of California.
Shahid, N., Naqvi, I. H., & Qaisar, S. B. (2015). Characteristics and classification of outlier detection techniques for wireless sensor networks in harsh environments: A survey. *Artificial Intelligence Review*, *43*(2), 193–228.
Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, *45*(4), 427–437.
Storey, M. V., Van der Gaag, B., & Burns, B. P. (2011). Advances in on-line drinking water quality monitoring and early warning systems. *Water Research*, *45*(2), 741–747.
Talagala, P. D., & Hyndman, R. J. (2019a). A feature-based procedure for detecting technical outliers in water-quality data: R package oddwater v.0.7.0 (Computer software manual): Zenodo.
Talagala, P. D., & Hyndman, R. J. (2019b). oddwater: Outlier detection in data from water-quality sensors. Retrieved from https://github.com/pridiltal/oddwater (R package).
Talagala, P. D., Hyndman, R. J., & Smith-Miles, K. (2019). Anomaly detection in high dimensional data. arXiv preprint arXiv:1908.04000.
Talagala, P. D., Hyndman, R. J., Smith-Miles, K., Kandanaarachchi, S., & Muñoz, M. A. (2019). Anomaly detection in streaming nonstationary temporal data. *Journal of Computational and Graphical Statistics*, 1–21. https://doi.org/10.1080/10618600.2019.1617160
Tang, J., Chen, Z., Fu, A. W.-C., & Cheung, D. W. (2002). Enhancing effectiveness of outlier detections for low density patterns. In *Pacific-Asia conference on knowledge discovery and data mining* (pp. 535–548). Berlin, Heidelberg: Springer.
Thottan, M., & Ji, C. (2003). Anomaly detection in IP networks. *IEEE Transactions on signal processing*, *51*(8), 2191–2204.
Tutmez, B., Hatipoglu, Z., & Kaymak, U. (2006). Modelling electrical conductivity of groundwater using an adaptive neuro-fuzzy inference system. *Computers & Geosciences*, *32*(4), 421–433.
Unwin, A. (2019). Multivariate outliers and the O3 plot. *Journal of Computational and Graphical Statistics*, *28*(3), 635–643.
Weissman, I. (1978). Estimation of parameters and large quantiles based on the k largest observations. *Journal of the American Statistical Association*, *73*(364), 812–815.
Wilkinson, L. (2018). Visualizing big data outliers through distributed aggregation. *IEEE Transactions on Visualization and Computer Graphics*, *24*(1), 256–266.
Yu, J. (2012). A Bayesian inference based two-stage support vector regression framework for soft sensor development in batch bioprocesses. *Computers & Chemical Engineering*, *41*, 134–144.
Zhang, K., Hutter, M., & Jin, H. (2009). A new local distance-based outlier detection approach for scattered real-world data. In *Pacific-Asia conference on knowledge discovery and data mining* (pp. 813–822). Berlin, Heidelberg: Springer.

Supporting Information for "A feature-based procedure for detecting technical outliers in water-quality data from in situ sensors"

Priyanga Dilini Talagala[1,2], Rob J. Hyndman[1,2], Catherine Leigh[1,3,4], Kerrie Mengersen[1,4], Kate Smith-Miles[1,5]

[1]ARC Centre of Excellence for Mathematics and Statistical Frontiers (ACEMS), Australia
[2]Department of Econometrics and Business Statistics, Monash University, Australia
[3]Institute for Future Environments, Science and Engineering Faculty, Queensland University of Technology, Australia
[4]School of Mathematical Sciences, Science and Engineering Faculty, Queensland University of Technology, Australia
[5]School of Mathematics and Statistics, University of Melbourne, Australia

Contents

Introduction

We considered the following outlier scoring techniques for the current work presented in this paper. The oddwater procedure can be easily updated with other unsupervised outlier scoring techniques.

Text S1.

NN-HD algorithm

This algorithm is inspired by the HDoutliers algorithm (Wilkinson, 2018) which is an unsupervised outlier detection algorithm that searches for outliers in high dimensional data assuming there is a large distance between outliers and the typical data. Nearest neighbor distances between points are used to detect outliers. However, variables with large variance can bring disproportional influence on Euclidean distance calculation. Therefore, the columns of the data sets are first normalized such that the data are bounded by the unit hyper-cube. The nearest neighbor distances are then calculated for each observation. In contrast to the implementation of HDoutliers algorithm available in the `HDoutliers` package (Fraley, 2018) our implementation available through the `oddwater` package now generates outlier scores instead of labels for each observation.

KNN-AGG and KNN-SUM algorithms

The NN-HD algorithm uses only nearest neighbor distances to detect outliers under the assumption that any outlying point present in the data set is isolated. For example, if there are two outlying points that are close to one another, but are far away from the rest of the valid data points, then the two outlying points become nearest neighbors to one another and give a small nearest neighbor distance for each outlying point. Because the NN-HD algorithm is dependent on the nearest neighbor distances, and the two outlying points do not show any significant deviation from other typical points with respect to nearest neighbor distance, the NN-HD algorithm now fails to detect these points as outliers.

Corresponding author: Priyanga Dilini Talagala, `dilini.talagala@monash.edu`

Following Angiulli and Pizzuti (2002), Madsen (2018) proposed two algorithms: aggregated k-nearest neighbor distance (KNN-AGG); and sum of distance of k-nearest neighbors (KNN-SUM) to overcome this limitation by incorporating k nearest neighbor distances for the outlier score calculation. The algorithms start by calculating the k nearest neighbor distances for each point. The k-dimensional tree (kd-tree) algorithm (Bentley, 1975) is used to identify the k nearest neighbors of each point in a fast and efficient manner. A weight is then calculated using the k nearest neighbor distances and the observations are ranked such that outliers are those points having the largest weights. For KNN-SUM, the weight is calculated by taking the summation of the distances to the k nearest neighbors. For KNN-AGG, the weight is calculated by taking a weighted sum of distances to k nearest neighbors, assigning nearest neighbors higher weight relative to the neighbors father apart.

LOF algorithm

The Local Outlier Factor (LOF) algorithm (Breunig et al., 2000) calculates an outlier score based on how isolated a point is with respect to its surrounding neighbors. Data points with a lower density than their surrounding points are identified as outliers. The local reachable density of a point is calculated by taking the inverse of the average readability distance based on the k (user defined) nearest neighbors. This density is then compared with the density of the corresponding nearest neighbors by taking the average of the ratio of the local reachability density of a given point and that of its nearest neighbors.

COF algorithm

One limitation of LOF is that it assumes that the outlying points are isolated and therefore fails to detect outlying clusters of points that share few outlying neighbors if k is not appropriately selected (Tang et al., 2002). This is known as a masking problem (Hadi, 1992), i.e. LOF assumes both low density and isolation to detect outliers. However, isolation can imply low density, but the reverse does not always hold. In general, low density outliers result from deviation from a high density region and an isolated outlier results from deviation from a connected dense pattern. Tang et al. (2002) addressed this problem by introducing a Connectivity-based Outlier Factor (COF) that compares the average chaining distances between points subject to outlier scoring and the average of that of its neighboring to their own k-distance neighbors.

INFLO algorithm

Detection of outliers is challenging when data sets contain adjacent multiple clusters with different density distributions (Jin et al., 2006). For example, if a point from a sparse cluster is close to a dense cluster, this could be misclassified as an outlier with respect to the local neighborhood as the density of the point could be derived from the dense cluster instead of the sparse cluster itself. This is another limitation of LOF (Breunig et al., 2000). The Influenced Outlierness (INFLO) algorithm (Jin et al., 2006) overcomes this problem by considering both the k nearest neighbors (KNNs) and reverse nearest neighbors (RNNs), which allows it to obtain a better estimation of the neighborhood's density distribution. The RNNs of an object, p for example, are essentially the objects that have p as one of their k nearest neighbors. Distinguish typical points from outlying points is helpful because they have no RNNs. To reduce the expensive cost incurred by searching a large number of KNNs and RNNs, the kd-tree algorithm was used during the search process.

LDOF algorithm

The Local Distance-based Outlier Factor (LDOF) algorithm (Zhang et al., 2009) also uses the relative location of a point to its nearest neighbors to determine the degree to which the point deviates from its neighborhood. LDOF computes the distance for an observation to its k-nearest neighbors and compares the distance with the average distances of the point's nearest neighbors. In contrast to LOF (Breunig et al., 2000), which uses local density, LDOF now uses relative distances to quantify the deviation of a point from its neighborhood system. One of the main differences between the two approaches (LDOF and LOF) is that LDOF represents the typical pattern of the data set by scattered points rather than crowded main clusters as in LOF (Zhang et al., 2009).

RKOF algorithm with Gaussian kernel

According to Gao, Hu, Zhang, Zhang, and Wu (2011), LOF is not accurate enough to detect outliers in complex and large data sets. Furthermore, the performance of LOF depends on the parameter k that determines the scale of the local neighborhood. The Robust Kernel-based Outlier Factor (RKOF) algorithm (Gao et al., 2011) tries to overcome these problems by incorporating variable kernel density estimates to address the first problem and weighted neighborhood density estimates to address the second problem. A Gaussian kernel with a bandwidth of $k-distance$ was used for density estimation. The two parameters: multiplication parameter for $k-distance$ of neighboring observations and sensivity parameter for $k-distance$ were set to 1 (default value given in Gao et al. (2011)).

References

Angiulli, F., & Pizzuti, C. (2002). Fast outlier detection in high dimensional spaces. In *European conference on principles of data mining and knowledge discovery* (pp. 15–27).

Bentley, J. L. (1975). Multidimensional binary search trees used for associative searching. *Communications of the ACM*, *18*(9), 509–517.

Breunig, M. M., Kriegel, H.-P., Ng, R. T., & Sander, J. (2000). Lof: identifying density-based local outliers. In *Acm sigmod record* (Vol. 29, pp. 93–104).

Fraley, C. (2018). Hdoutliers: Leland wilkinson's algorithm for detecting multidimensional outliers [Computer software manual]. Retrieved from `https://CRAN.R-project.org/package=HDoutliers` (R package version 1.0)

Gao, J., Hu, W., Zhang, Z. M., Zhang, X., & Wu, O. (2011). Rkof: robust kernel-based local outlier detection. In *Pacific-asia conference on knowledge discovery and data mining* (pp. 270–283).

Hadi, A. S. (1992). Identifying multiple outliers in multivariate data. *Journal of the Royal Statistical Society. Series B (Methodological)*, 761–771.

Jin, W., Tung, A. K., Han, J., & Wang, W. (2006). Ranking outliers using symmetric neighborhood relationship. In *Pacific-asia conference on knowledge discovery and data mining* (pp. 577–593).

Madsen, J. H. (2018). Ddoutlier: Distance and density-based outlier detection [Computer software manual]. Retrieved from `https://CRAN.R-project.org/package=DDoutlier` (R package version 0.1.0)

Tang, J., Chen, Z., Fu, A. W.-C., & Cheung, D. W. (2002). Enhancing effectiveness of outlier detections for low density patterns. In *Pacific-asia conference on knowledge discovery and data mining* (pp. 535–548).

Wilkinson, L. (2018). Visualizing big data outliers through distributed aggregation.

IEEE transactions on visualization and computer graphics, *24*(1), 256–266.

Zhang, K., Hutter, M., & Jin, H. (2009). A new local distance-based outlier detection approach for scattered real-world data. In *Pacific-asia conference on knowledge discovery and data mining* (pp. 813–822).

Chapter 5

A Framework for Automated Anomaly Detection in High Frequency Water-Quality Data From *in situ* Sensors

This article is published in the *Science of the Total Environment*. The work is based on the collaborative research project carried out with the Queensland University of Technology and the Queensland Department of Environment and Science, Great Barrier Reef Catchment Loads Monitoring Program from April to July 2018.

Science of the Total Environment 664 (2019) 885–898

Contents lists available at ScienceDirect

Science of the Total Environment

journal homepage: www.elsevier.com/locate/scitotenv

A framework for automated anomaly detection in high frequency water-quality data from in situ sensors

Catherine Leigh [a,b,c,*], Omar Alsibai [a,b], Rob J. Hyndman [a,d], Sevvandi Kandanaarachchi [a,d], Olivia C. King [e], James M. McGree [a,c], Catherine Neelamraju [e], Jennifer Strauss [e], Priyanga Dilini Talagala [a,d], Ryan D.R. Turner [e], Kerrie Mengersen [a,c], Erin E. Peterson [a,b,c]

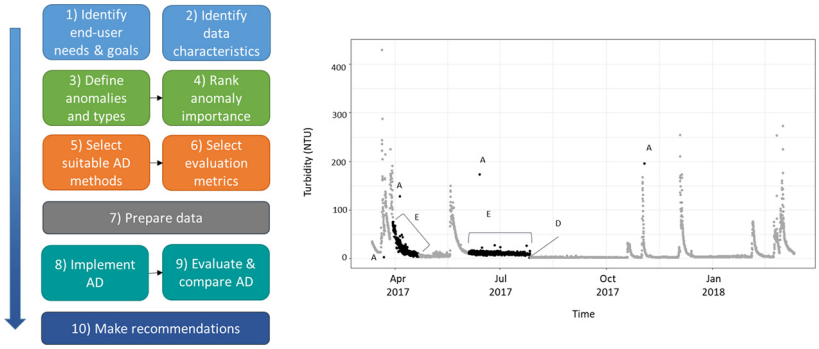
[a] ARC Centre of Excellence for Mathematical & Statistical Frontiers (ACEMS), Australia
[b] Institute for Future Environments, Queensland University of Technology, Brisbane, Queensland, Australia
[c] School of Mathematical Sciences, Science and Engineering Faculty, Queensland University of Technology, Brisbane, Queensland, Australia
[d] Department of Econometrics and Business Statistics, Monash University, Clayton, Victoria, Australia
[e] Water Quality and Investigations, Department of Environment and Science, Dutton Park, Queensland, Australia

HIGHLIGHTS

- High frequency water-quality data requires automated anomaly detection (AD).
- Rule-based methods detected all missing, out-of-range and impossible values.
- Regression and feature-based methods detected sudden spikes and level shifts well.
- High false negative rates were associated with other types of anomalies, e.g. drift.
- Our transferable framework selects and compares AD methods for end-user needs.

GRAPHICAL ABSTRACT

The ten-step Anomaly Detection (AD) framework for high frequency water-quality data, which includes ranking the importance of different anomaly types (e.g. sudden spikes A, sudden shifts D, anomalously high variation type E), based on end-user needs and data characteristics, to inform algorithm choice, implementation and performance evaluation. Framework numbers indicate the order of steps taken. Arrows indicate directions of influence between steps.



ARTICLE INFO

Article history:
Received 22 October 2018
Received in revised form 4 February 2019
Accepted 5 February 2019
Available online 06 February 2019

Editor: Patricia Holden

Keywords:
Big data
Forecasting

ABSTRACT

Monitoring the water quality of rivers is increasingly conducted using automated in situ sensors, enabling timelier identification of unexpected values or trends. However, the data are confounded by anomalies caused by technical issues, for which the volume and velocity of data preclude manual detection. We present a framework for automated anomaly detection in high-frequency water-quality data from in situ sensors, using turbidity, conductivity and river level data collected from rivers flowing into the Great Barrier Reef. After identifying end-user needs and defining anomalies, we ranked anomaly importance and selected suitable detection methods. High priority anomalies included sudden isolated spikes and level shifts, most of which were classified correctly by regression-based methods such as autoregressive integrated moving average models. However, incorporation of multiple water-quality variables as covariates reduced performance due to complex relationships among variables. Classifications of drift and periods of anomalously low or high variability were more often correct

Abbreviations: AD, anomaly detection; ADAM, anomaly detection and mitigation; ARIMA, autoregressive integrated moving average; FN, false negative; FP, false positive; PI, prediction interval; PR, Pioneer River; RegARIMA, multivariate regression with ARIMA errors; SC, Sandy Creek; TN, true negative; TP, true positive.

* Corresponding author at: ARC Centre of Excellence for Mathematical & Statistical Frontiers (ACEMS), Australia.
E-mail address: catherine.leigh@qut.edu.au (C. Leigh).

https://doi.org/10.1016/j.scitotenv.2019.02.085

Near-real time
Quality control and assurance
River
Time series

when we applied mitigation, which replaces anomalous measurements with forecasts for further forecasting, but this inflated false positive rates. Feature-based methods also performed well on high priority anomalies and were similarly less proficient at detecting lower priority anomalies, resulting in high false negative rates. Unlike regression-based methods, however, all feature-based methods produced low false positive rates and have the benefit of not requiring training or optimization. Rule-based methods successfully detected a subset of lower priority anomalies, specifically impossible values and missing observations. We therefore suggest that a combination of methods will provide optimal performance in terms of correct anomaly detection, whilst minimizing false detection rates. Furthermore, our framework emphasizes the importance of communication between end-users and anomaly detection developers for optimal outcomes with respect to both detection performance and end-user application. To this end, our framework has high transferability to other types of high frequency time-series data and anomaly detection applications.

1. Introduction

Clean water is a United Nations Sustainable Development Goal as well as a major concern in many developed areas. Monitoring the quality of water in the world's rivers relies predominantly on manual collection of water-quality samples at low frequencies (e.g. monthly). These discrete samples are analysed in laboratories to provide estimates of the concentrations of ecologically important constituents such as sediments and nutrients. This requires considerable time and money, and the resulting data are typically sparse in space and time. Fortunately, other properties of water, such as turbidity and conductivity, can be measured semi-continuously by readily available, low-cost, automated in situ sensors, and show promise as surrogates of sediment and nutrient concentrations in rivers (Jones et al., 2011; Slaets et al., 2014). Nevertheless, technical issues in sensor monitoring equipment can occur, for example, when battery power is low or sensors drift over time due to biofouling of the probes, or due to errors in calibration. These issues can lead to errors in water-quality measurements, which we define herein as anomalies. Such anomalies can be important to detect because they can confound the assessment or identification of true changes in water quality.

Notwithstanding technical errors, another issue that mitigates the potential advantages of using in situ sensor data is the production of high-frequency water-quality data in near-real time (i.e. data streaming). This high velocity, high volume data creates problems for quality control and assurance, given that manual checking, labelling and correction of each observation is unfeasible (Hill and Minsker, 2010; Horsburgh et al., 2015). We therefore need to develop robust methods for automatic anomaly detection (AD) before water-quality data from in situ sensors can be used with confidence for water-quality visualization, analysis and reporting.

Here, our objective was to develop a ten-step AD framework to implement and compare a suite of AD methods for high-frequency water-quality data measured by in situ sensors (Fig. 1). We demonstrate this framework using a real-world case study where turbidity, conductivity and river level data were measured by automated in situ sensors in rivers flowing into the Great Barrier Reef lagoon of northeast Australia. Anomalies were defined as any water-quality observations that were affected by technical errors in the sensor equipment; in other words, not due to real ecological patterns and processes occurring within the rivers and watersheds being monitored. We focussed on AD in turbidity and conductivity data because these properties of river water are typically more stable through time than other properties such as dissolved oxygen and temperature, which fluctuate daily as well as seasonally. Turbidity and conductivity were also the target water-quality variables for the end-user in our case study, described in Sections 2.1–2.2. We present this framework below and discuss the results of AD for high-frequency water-quality data from automated in situ sensors, with a view to providing insight on broader applications and future directions.

2. Methods

We describe below the method components of the AD framework (Steps 1 to 8; Fig. 1) from identifying end-user needs and anomaly types and priorities through to selecting suitable analytical methods of AD.

2.1. Identify end-user needs and goals (Step 1)

Identifying the needs and goals of the end-user is a key step in the AD framework because this helps determine which types of anomalies will be most important to detect and thus the most suitable AD methods (Fig. 1, Table 1). For this case study, several discussions were held between the end-user (an environmental agency concerned with water quality monitoring and management), statisticians and freshwater scientists prior to commencing analysis. The foremost, short-term need of the environmental agency was to produce 'smart' graphical outputs of the streaming water-quality data from in situ sensors for visualization

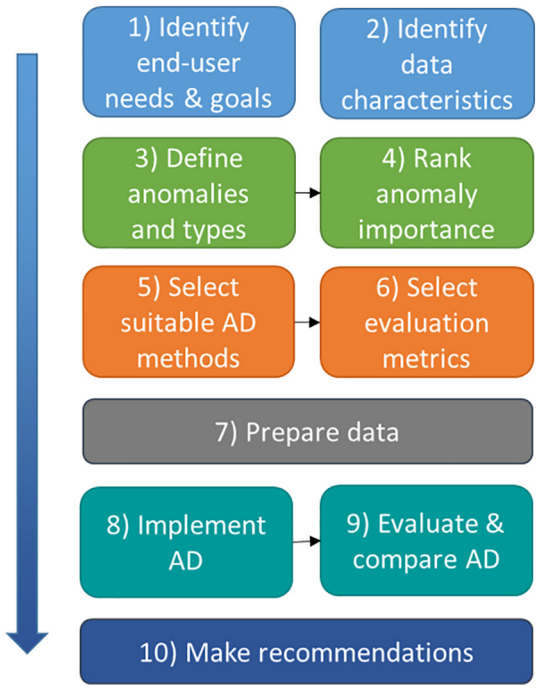


Fig. 1. The ten-step Anomaly Detection (AD) framework for high frequency water-quality data, which includes defining and ranking the importance of different types of anomalies, based on end-user needs and data characteristics, to inform algorithm choice, implementation, performance evaluation and resultant recommendations. Numbers indicate the order of steps taken. Arrows indicate directions of influence between steps.

Table 1
Types of anomalies likely encountered in in situ sensor-generated water-quality time series, along with the importance ranking of each type with respect to the priority end-user goal in this case study (i.e. time series visualization), and relevance to potential end-users.

Anomaly type	Type code (Class[b])	Description	Examples in the literature and/or alternative terminology	Importance ranking (with respect to time series visualization in this case study)	Potential end-users and applications[c]
Large sudden spike	A[a] (1)	Anomalous value is isolated and 'much' higher or lower than surrounding data, and the spike occurs in a very short window of time (e.g. only one data point is anomalously high or low).	Point or collective anomaly (Goldstein and Uchida, 2016)	First priority (at any point in the time series)	Management, monitoring and compliance; Policy and decision makers; Public; Data managers; Sensor maintenance technicians
Low variability/ persistent values	B (3)	Values constant though time or with very minimal variation compared with that expected	Data value persistence (Horsburgh et al., 2015); collective anomaly (Chandola et al., 2009)	Second priority (especially during event flow)	Data managers; Sensor maintenance technicians
Constant offset (e.g. calibration error)	C (3)	Values are in error by some constant. Likely related to/seen before and/or after sudden shifts	Incorrect offset or calibration (Horsburgh et al., 2015)	Third priority	Data managers; Sensor maintenance technicians
Sudden shifts	D (1)	Values suddenly shift to a new range (higher or lower than previous range)	Level shifts (Tsay, 1988)	Equal third priority (especially when shift is considered large)	Management, monitoring and compliance; Policy and decision makers; Public; Data managers; Sensor maintenance technicians
High variability	E (3)	Values oscillate considerably over short time periods (more than expected during natural daily cycles or events)	Variance change (Tsay, 1988); collective anomaly (Chandola et al., 2009)	Fourth priority	Data managers; Sensor maintenance technicians
Impossible values	F (2)	Values impossible or highly unlikely for that water-quality variable (e.g. negative values for all, conductivity values nearing or at zero ('too fresh'))	Out of range values (Horsburgh et al., 2015)	Important, but should be detected easily (e.g. using a simple rule)	Sensor manufacturers; Statisticians; Data managers; Sensor maintenance technicians
Out-of-sensor-range values	G (2)	Values that the sensors are incapable of detecting (outside of their detection range). Some of these anomalies may be first captured under type F (impossible values)	Not distinguished from type F by Horsburgh et al. (2015)	Important, but should be detected easily (e.g. using a simple rule)	Sensor manufacturers; Statisticians; Data managers; Sensor maintenance technicians
Drift	H (3)	Gradual change in values in positive or negative direction	Sensor drift (Horsburgh et al., 2015); collective anomaly (Chandola et al., 2009)	Comparatively low priority (most likely observed in turbidity), but important to flag as being a possible occurrence of an anomaly e.g. when gradual increase or decrease occurs before a sudden shift	Sensor manufacturers; Data managers; Sensor maintenance technicians
Clusters of spikes	I[a] (1)	Multiple spikes in a short period of time	Micro cluster (Goldstein and Uchida, 2016); collective anomaly (Chandola et al., 2009)	Low priority (isolated spikes much more important to detect)	Management, monitoring and compliance; Policy and decision makers; Public; Data managers; Sensor maintenance technicians
Small sudden spike	J[a] (1)	Anomalous value is 'somewhat' higher or lower than surrounding data, and the spike occurs in a very short window of time (e.g. only one data point is anomalously high or low)	Point anomaly (Goldstein and Uchida, 2016)	Very low priority	Data managers; Sensor maintenance technicians
Missing values	K (2)	Gaps in time series (i.e. greater than the set frequency of measurement)	Skipped or no-data values (Horsburgh et al., 2015)	Undetermined	Data managers; Sensor maintenance technicians; Sensor manufacturers; Statisticians; Policy and decision makers

[a] Spikes may be in the positive or negative direction with respect to surrounding data (i.e. can include a sudden isolated decrease and/or a sudden isolated increase in value).
[b] Classes of anomalies, as defined in this paper: (1) involve a sudden change in value from the previous observation, (2) are detectable by automated classification rules, (3) likely require user intervention to identify observations as anomalous.
[c] Monitoring, management and compliance: agencies, industries and landholders etc. concerned with water quality monitoring, management and compliance checking – summary statistics such as means are strongly influenced by such anomalies; Policy and decision makers – to limit use of incorrect data and for reporting purposes; Public – to avoid false warning of water quality breaches; Data managers – for quality control and assurance and to increase confidence in the data by reporting the presence of such anomalies; Sensor maintenance technicians – to ensure timely and correct calibration and maintenance of equipment; Sensor manufacturers – to improve performance, e.g. extend battery life, improve wiper quality to further minimize biofouling; Statisticians – for AD methods to better detect other non-trivial anomaly types and/or for methods requiring regular and frequent observations.

in near-real time (Table 1). Visualization of streaming water-quality data helps to engender confidence in those data, and this means that potentially anomalous water-quality observations need to be identified and labelled as such, in near-real time. The longer-term goals of the end-user, beyond the specific scope of this case study, were to provide complete quality control and assurance of the data; flagging potential anomalies in near-real time, potentially with automated correction, and ultimately to use the corrected data to estimate sediment and nutrient concentrations in rivers in near-real time. Resultant data can then be used for accurate load estimation across multiple temporal scales. For other end-users, for example, the public, priority goals may include on-line and real-time warning of water quality breaches associated with real events (rather than technical anomalies). Such events may have serious economic and public health consequences in practice, affecting commercial operations (e.g. fisheries and aquaculture) and recreational sites (e.g. Rabinovici et al., 2004).

2.2. Identify data characteristics (Step 2)

Temporal characteristics of the time series data on which AD is performed play a role in determining the types of methods most suitable to use (Fig. 1). Here, we used water-quality data from in situ sensors deployed in rivers of tropical northeast Australia that flow into the Great Barrier Reef lagoon. The rivers of interest are located in the Mackay Whitsunday region, east of the Great Dividing Range in Queensland, Australia. This region is characterized by a highly seasonal climate, experiencing higher rainfall and air temperatures in the 'wet' season (typically occurring between December and April and associated with event flows in rivers) and lower rainfall and air temperatures in the 'dry' season (associated with low to zero flows in rivers; Brodie, 2004). These characteristics affect the patterns of water quality in these rivers through time.

We focussed on two rivers in the study region: Pioneer River and Sandy Creek. The upper reaches of Pioneer River flow predominantly through National or State Parks, with its middle and lower reaches flowing through land dominated by sugarcane farming. Sandy Creek is a low-lying coastal-plain stream, south of the Pioneer River, with a similar land-use and land-cover profile to that of the lower Pioneer River. Two study sites, one on Pioneer River and one on Sandy Creek (PR and SC, respectively), are in freshwater reaches and their monitored catchment areas are 1466 km^2 and 326 km^2, respectively.

Automated water-quality sensors (YSI EXO2 Sondes with YSI Smart Sensors attached) have been installed at each site, housed in flow cells in water-quality monitoring stations on riverbanks. At each site, a pumping system is used to transport water regularly from the river up to the flow cell, approximately every hour or hour and a half, for the sensors to measure and record turbidity (NTU) and electrical conductivity at 25 °C (conductivity; μS/cm). All equipment undergo regular maintenance and calibration, with sensors calibrated and equipment checked approximately every 6 weeks following manufacturer guidelines. Sensors are equipped with wipers to minimize biofouling. Pre-wet season maintenance, e.g. cleaning samplers and drainage lines from the flow cell, is performed annually.

Turbidity is an optical property of water that reflects its level of clarity, which declines as the concentrations of abiotic and biotic suspended particles that absorb and scatter light increase. Turbidity thus tends to increase rapidly during runoff events when waters contain high concentrations of suspended particles eroded from the land and upstream river channels. When waters concentrate during times of low or zero flow, turbidity may increase gradually through time. Similarly, conductivity, which reflects the concentration of ions including bioavailable nutrients such as nitrate and phosphate in the water, also tends to increase during periods of low and zero flow, but can decrease rapidly with new inputs of fresh water. Measurements of turbidity and conductivity may be taken more frequently during event flows to capture the increased range of values observed during such times; however, the relationships among river level, turbidity and conductivity are complex (Fig. S1). River level (i.e. height in meters from the riverbed to the water surface; level, m) is recorded by sensors at each site every 10 min; we linearly interpolated these data to provide time-matched observations of level for each turbidity and conductivity observation. Although we did not perform anomaly detection on the river level data, we examined its relationship with the turbidity and conductivity data to provide insight into the water-quality dynamics occurring at both study sites (Fig. S1). The time series data from the in situ sensors were available from 12 March 2017 to 12 March 2018 at both sites, totalling 6280 and 5402 observations at PR and SC, respectively (Figs. S2–S3).

2.3. Define anomalies and their types (Step 3)

A clear definition of what constitutes an anomaly, relevant to the data and end-user requirements, is needed prior to commencing detection (Fig. 1). Several definitions of anomalies exist, each differing in specificity. In general, they are considered to (i) differ from the norm with respect to their features, and (ii) be rare in comparison with the non-anomalous observations in a dataset (Goldstein and Uchida, 2016). As mentioned, we defined an anomaly here as any water-quality datum or set of data that was due to a technical error in the in situ sensor equipment. For example, a real datum might include a rare, high-magnitude value of turbidity associated with heavy, erosive local rainfall and an ensuing high-flow event, whereas an anomaly might be a similarly high datum but one that is beyond the range of detection by the sensor.

Once 'anomaly' is defined, the different types of anomalies likely to be present in the time series data of interest must be defined and identified. We defined the different types of anomalies likely to occur in the water-quality data, in consultation with the end-user in this study, as: sudden spikes (large A, small J), low variability including persistent values (B), constant offsets (C), sudden shifts (D), high variability (E), impossible values (F), out-of-sensor-range values (G), drift (H), clusters of spikes (I), missing values (K) and other, untrustworthy (L; not described by types A-K) (Table 1, Fig. 2). Some of these types have been described elsewhere for high frequency water-quality data, using the same or different terminology (e.g. Horsburgh et al., 2015), whilst other types were identified as more relevant to the specific characteristics of the data we were analysing (e.g. periods of anomalously high variation; Table 1). Other terms, such as local and global anomalies, have been used to describe anomalies in other contexts. We do not use these other terms here, chiefly because they do not adequately differentiate between the relevant types of anomalies we defined. For example, local anomalies, as defined by Goldstein and Uchida (2016), are only anomalous when compared with their immediate neighbourhood of data values. These may include large or small sudden spikes, values that are anomalously different in magnitude to that of data at neighbouring time steps. Global anomalies, on the other hand, are anomalously different to the majority of other data points, regardless of time (Goldstein and Uchida, 2016). Contextual anomalies describe data that appear anomalous only when context (e.g. season) is taken into account, otherwise appearing 'normal' (Goldstein and Uchida, 2016). For example, a high value of river level may be non-anomalous in the wet season, but could be anomalous within the context of the dry season. Contextual anomalies may, for example, include some anomalies identified here as type L (other, untrustworthy data). Types B, E, H and I anomalies may be referred to elsewhere as collective anomalies, i.e. collections of anomalous data points (Chandola et al., 2009). We additionally labelled the first observation after an extended period of missing data (i.e. no observations for >180 min) to identify it as an anomaly type K (see also Section 2.5.1).

We grouped the anomaly types into three general classes (Table 1). Class 1 included anomalies described by a sudden change in value from the previous observation (types A, D, I, and J). Class 2 included those anomaly types that should be detectable by simple, hard-coded

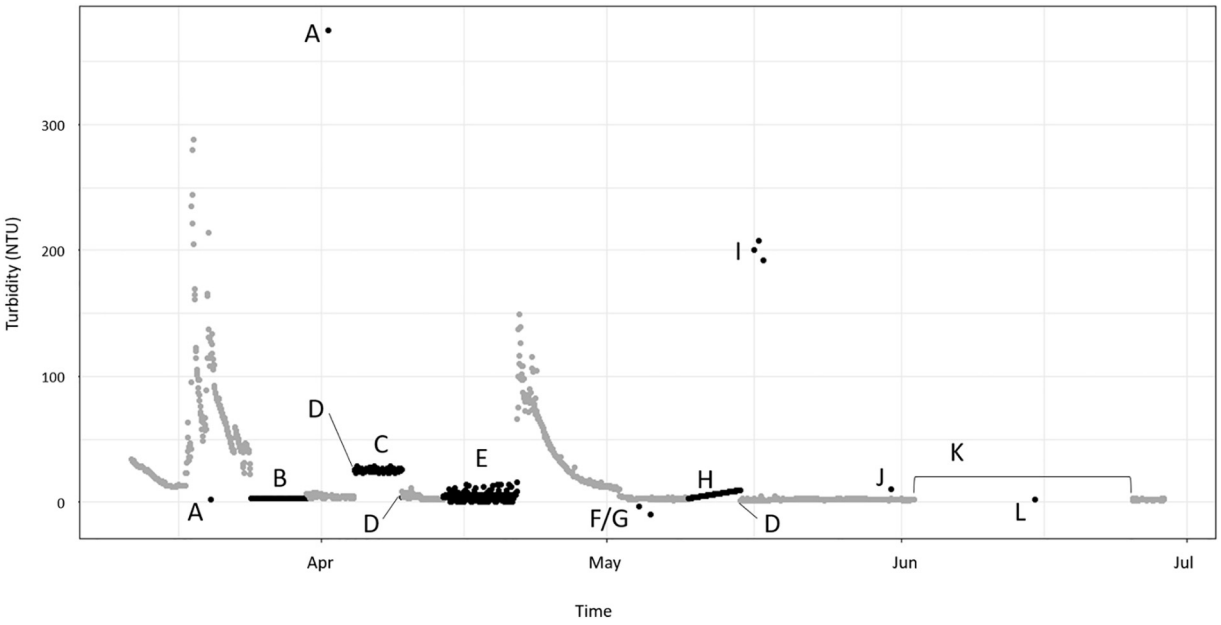


Fig. 2. Example of a turbidity (NTU) time series featuring both normal observations (dark grey points) and anomalies (black points; labelled A-L according to Table 1).

classification rules, such as measurements outside the detectable range of the sensor (types F, G and K), whereas Class 3 anomalies may require user intervention post hoc (i.e. after data collection rather than in real time) to confirm observations as anomalous or otherwise in combination with automated detection (types B, C, E, H and L). Finally, we noted the times at which sensor maintenance activities such as probe swapping for calibration purposes were performed, to flag when anomalies might be likely to occur and provide causal insight about anomaly generation (Figs. S2–S3).

We visually examined the water-quality time series data in consultation with the end-user. The potential anomalies in each time series at each site were identified and labelled along with their types based on expert knowledge of riverine water-quality dynamics and the particular sites and watersheds of interest. The labelled anomalies were used in steps 8–9 to implement AD and assess its performance.

2.4. Rank anomaly types by importance (Step 4)

The importance ranking for anomaly types is based on the potential impact each type may have if it were to go undetected, with respect to end-user needs and goals. This ensures that the end-user can effectively assess the ability of the AD methods to identify the most important anomalies. For example, one method may detect the same amount of anomalies as another; whilst the first method identifies anomalous high-magnitude values in a turbidity time series, the second method instead identifies minimally negative (impossible) values during periods of low turbidity only. If the end-user deems the former type of anomaly as more important to detect, then this would affect the evaluation of which AD method performs best and is most suitable. The rationale for the ranking might be that high-magnitude anomalies falsely indicate a breach of water-quality guidelines, whereas the change in turbidity caused by the negative readings may be negligible in the context of the period in which they occurred.

Here, we liaised with the end-user (in this case, an environmental agency concerned with water management and monitoring, see Section 2.1) to rank the importance of the different anomaly types identified as per Section 2.3 (Table 1). Their first priority was to identify large sudden spikes (Type A, Class 1) given that the short-term aim of anomaly detection was time series visualization. Sudden shifts (Type D, Class 1), calibration offsets (Type C, Class 3) and changes in variance (Types B and E, Class 3) were also deemed important, ranking second to fourth in priority, with types C and D both ranked third in place (Table 1).

2.5. Select suitable methods of anomaly detection (Step 5)

As outlined in Step 2 (Section 2.2), characteristics of the data on which AD is performed play a role in determining the most suitable AD methods, taking the end-user needs into account. Time series data are typically nonstationary, such that statistical parameters of the data (e.g. the mean and standard deviation) change with time. Furthermore, the production of high-frequency water-quality data from in situ sensors in near-real time creates 'big data', i.e. high-volume, high-velocity and high-variety information (Gandomi and Haider, 2015). This may be problematic for certain AD methods such as those developed for or typically applied to relatively small batches of pre-collected (historical) data (Liu et al., 2015).

We reviewed and compared the different AD methods used for water quality and time series data as described in the literature to identify those that are, or could be made, suitable for analysing near-real time and nonstationary data streams (Table S1). This included automated classification rules as well as several regression and feature-space based methods. Many of these methods are well documented and freely available software is available to implement them. Thus, they serve as suitable benchmarks for new anomaly detection methods that may be developed in the future. We chose to implement a suite of these methods because different algorithms are likely to detect certain types of anomalies (e.g. priority anomalies like large sudden spikes; Table 1) better than others.

Although we also considered physical-process based models for AD in water-quality time series (e.g. Moatar et al., 2001; Table S1), we did not explore them further here. Variation in water-quality patterns through time in rivers, and the multiple interactions within and between water-quality variables and the broader environment creates complexities and uncertainties that can make development of such models challenging and limit their transferability (e.g. Cox, 2003), particularly in the context of streaming data. Likewise, we did not explore dynamic Bayesian networks or hidden Markov models (Table S1). Whilst both methods show potential in the context of streaming time series data (Hill et al., 2009; Li et al., 2017), their application in this context is relatively new with limited existing software for implementation using water-quality data.

2.5.1. Automated classification rules

Perhaps the simplest way to detect and classify anomalies such as impossible, out-of-sensor-range and missing values (Class 2: type F, G and K, respectively) is to develop rules that can be automated and applied to the streaming data in near-real time in combination with data-driven approaches such as regression and feature-based AD (see Sections 2.5.2–2.5.3). For instance, negative values are impossible for turbidity and a simple rule (e.g. a 'range test'; Fiebrich et al., 2010) could therefore be set to classify any negative turbidity observation as an anomaly. Here, we implemented 'if-then' statements to detect and classify Class 2 anomalies. The first statement classified type K anomalies, using an end-user defined period as the maximum allowable time between two consecutive observations before the second observation is classed as a K, indicating that it occurred after a period of missing observations. Here we defined the maximum allowable threshold as 180 min (3 h); however, this definition will vary according to end-user needs and the frequency of the in situ sensor data. We next identified type F anomalies (i.e. impossible values); if a turbidity or conductivity observation was negative, then it was classed as an anomaly. Furthermore, if any turbidity or conductivity observation was zero, then it was likewise classed as an anomaly because completely clear river water containing zero positive or negative ions is unrealistic. The if-then statements used to detect type G anomalies were based on range tests defined by sensor specifications for each water-quality variable.

2.5.2. Regression-based methods

The regression-based approach to AD in time series has several advantages, including (for some methods) the ability to deal with nonstationarity and provide near-real time support (Table S1). Furthermore, information from single or multiple water-quality variables as well as previous measurements can be taken into account, which makes these methods useful in the context of AD for streaming water-quality data. Most regression-based methods used for AD are semi-supervised (Table S1); the models are trained to learn the 'normal' (i.e. non-anomalous, typical) behaviour in a time series and, theoretically, should then detect any non-normal (i.e. anomalous) behaviour, regardless of the underlying cause.

To perform AD, the regression-based methods are used to generate a prediction, or forecast, with an associated measure of uncertainty at the next time point. The prediction intervals should ideally account for uncertainty associated with the model, model parameter values and the behaviour of future data, although in practice often only the model uncertainty is taken into account (Hyndman and Athanasopoulos, 2018). If the one-step-ahead observation does not fall within the prediction interval, it is classified as an anomaly.

The general form for regression-based methods can be written as:

$$x_t = \beta' Z_t + \eta_t$$

$$\eta_t = ARIMA(p, d, q)$$

where x_t is the observation at time t, β' is a vector of regression coefficients, and Z_t is a vector of covariates. Thus, the errors from the regression model may be serially correlated, and we model this correlation structure using an ARIMA model. ARIMA models are discussed further below, and in detail in Hyndman and Athanasopoulos (2018), and can be thought of as a nonlinear regression against past observations. We assume the ARIMA model errors are uncorrelated in time, and normally distributed with zero mean, and we denote this by $\varepsilon_t \sim N(0, \sigma)$.

We let $\tilde{x}_{t+1}$ denote the one-step forecast of x_{t+1} made at time t. To compute these forecasts, we add $\beta' Z_{t+1}$ to the forecasts from the ARIMA model.

After forecasting, observations are classified as anomalies, or not, based on the specified prediction interval. There is no training involved in this step. Here, we constructed a 100(1-α)% prediction interval (*PI*) for the one-step-ahead prediction (the forecast observation at time $t + 1$) as:

$$PI_{t+1} = \tilde{x}_{t+1} \pm t_{\alpha/2, T-k} \times s$$

where T is the size of the training dataset, k is the number of parameters in the model, $t_{\alpha/2,\ T-k}$ is the $\alpha/2$ quantile of a t-distribution with T - k degrees of freedom, and s is the square root of the mean of the squared ARIMA residuals in the training dataset.

The *PI* defines the range of 'normal' (i.e. non-anomalous) one-step-ahead predictions. The choice of significance level therefore affects the number of false positives produced. There were relatively few labelled anomalies in our time series data, especially for certain water-quality variables and anomaly types (Table 2). We therefore used a 99% prediction interval ($\alpha = 0.01$) to effectively limit the probability of false positives to 1%.

We implemented the following set of regression-based models, based on the general form, to detect anomalies in the turbidity and conductivity time series: (i) naïve prediction, (ii) linear autoregression, (iii) ARIMA models, and (iv) multivariate linear regression with ARIMA errors (RegARIMA).

Naïve prediction is a regression-based method that uses the most recent observation as the one-step-ahead forecast:

$$\tilde{x}_{t+1} = x_t$$

In the notation of our general model, $\beta = Z_t = 0$ and $\eta_t = ARIMA(0,1,0)$. The method assumes the one-step-ahead forecast depends only on the previous observation, therefore the only parameter to estimate is s, the square root of the mean squared residuals, where the residuals in this case are the differences between consecutive observations. Naïve prediction therefore does not require stationarity in the mean of the time series (Table S1).

Table 2
Number of anomalous observations identified according to type, class and water-quality variable at Pioneer River (PR) and Sandy Creek (SC). Number of instances of Class 3 anomalies that comprise multiple contiguous observations, and where relevant their neighbouring anomaly types, in parentheses.

Site	Anomaly type and class	Turbidity	Conductivity	Level	Total
PR	A (Class 1)	1	2	0	3
	D (Class 1)	3	2	0	5
	F (Class 2)	0	34	0	34
	H (Class 3)	0	397 (1 instance, before a D)	0	397
	J (Class 1)	5	0	2	7
	K (Class 2)	4	4	4	12
	L (Class 3)	718 (1 instance, between two Ds)	80 (2 instances, the first after a D, the second between Ks)	0	798
	Class 1	9	4	2	15
	Class 2	4	38	4	46
	Class 3	718	477	0	1195
	Total (out of 6280 observations)	731	519	6	1256
SC	A (Class 1)	4	1	0	5
	D (Class 1)	1	0	0	1
	E (Class 3)	914 (2 instances, the second before a D)	0	0	914
	F (Class 2)	0	0	1	1
	K (Class 2)	1	1	1	3
	Class 1	5	1	0	6
	Class 2	1	1	2	4
	Class 3	914	0	0	914
	Total (out of 5402 observations)	920	2	2	924

Linear autoregression (Box and Jenkins, 1970) differs from naïve prediction because it gives a forecast that is a linear combination of the p previous observations, rather than just the single previous observation:

$$\tilde{x}_{t+1} = c + \sum_{i=1}^{p} \phi_i x_{t-i}$$

where the constant c and the set $\{\phi_1, \phi_2, \ldots, \phi_p\}$ are model parameters estimated from the training data. In the notation of our general model, $c = \beta$, $Z_t = 1$ and $\eta_t = ARIMA(p,0,0)$. We used the partial autocorrelation function (PACF) to select the optimal value of p for the linear autoregression models (Tsay, 1989).

The ARIMA(p,d,q) model introduced by Box and Jenkins (1970) is more generalised than naïve prediction or linear autoregression models and includes autoregressive (p), differencing (d) and moving average (q) components (i.e. the succession of averages calculated from successive segments of the time series). Here, p determines the number of previous observations (time lags) in the autoregressive model, d determines the number of differences between observations to use, and q determines the number of moving average terms (see also Hyndman and Athanasopoulos, 2018). ARIMA models can handle stationary as well as nonstationary time series by adding a differencing component, i.e. using $d > 0$. To decide on the optimal value of the p, d and q ARIMA components, we used an automated procedure, based on the Akaike information criterion (AIC; Akaike, 1974); minimizing the AIC is asymptotically equivalent to using cross-validation (Hyndman and Athanasopoulos, 2018).

Finally, RegARIMA models, also known as dynamic regression models, are a combination of ARIMA time series modelling and multivariate regression (Hyndman and Athanasopoulos, 2018), where multivariate regression uses information from multiple water-quality variables for forecasting the one-step-ahead prediction:

$$\tilde{x}_{t+1} = \beta_0 + \sum_{i=1}^{k} \beta_i z_{i,t+1} + \tilde{\eta}_{t+1}$$

where $z_{i,\,t+1}$ represents variable i from the set of variables $\{1, \ldots, k\}$ at some time $t + 1$. In this way, information from multiple variables are included in the model in addition to information provided by previous observations. Here we included turbidity and river level, or conductivity and river level, in the multiple regression component of the ARIMA model to forecast the one-step-ahead conductivity, or turbidity observations, respectively, using the AIC to select the best p, d and q parameters as per ARIMA above.

For all of the above methods we investigated assumptions of the models by conducting Box-Ljung portmanteau tests to assess stationarity in the mean (Ljung and Box, 1978) and producing diagnostics plots to visually assess stationarity in variance.

One additional approach to AD within the regression-based suite of methods, applied to water-quality time series by Hill and Minsker (2010), uses anomaly mitigation (i.e. correction) during forecasting and classification. Essentially this anomaly detection and mitigation (ADAM) approach uses forecasts instead of actual observations, when detected as anomalous, to forecast the subsequent one-step-ahead observation. ADAM therefore has the potential to change the regression forecasting performance. After implementing each of the four regression-based methods outlined above on the time series data, we re-implemented them using the ADAM approach.

2.5.3. Feature-based methods

The feature-based approach to anomaly detection can make use of multiple time series to identify observations that deviate by distance or density from the majority of data in high dimensional 'feature space' (e.g. Talagala et al., 2018; Wilkinson, 2018). In the initial phase, transformations (e.g. log or differencing transformations) are applied to multiple time series to highlight different anomalies, such as sudden spikes and shifts. Different unsupervised anomaly detection methods are then applied to the high dimensional data space constructed by the transformed series to classify the anomalies. Because feature-based methods take the correlation structure of multiple water-quality variables into account, the anomaly classifications have a probabilistic interpretation. In other words, the anomalous threshold is not a user-defined parameter, but is instead determined by the data using probability theory. This increases the generalisability of such methods across different applications. Feature-based methods are computationally efficient and as such are suitable for analysing big data in near-real time. In addition, they are unsupervised, data-driven approaches and therefore do not require training (Table S1). Here, we implemented HDoutliers (Wilkinson, 2018), aggregated k-nearest neighbour (kNN-agg; Angiulli and Pizzuti, 2002; Madsen, 2018) and summed k-nearest neighbour AD (kNN-sum; Madsen, 2018) on one set of multivariate data for each site: the turbidity and conductivity time series.

The HDoutliers algorithm proposed by Wilkinson (2018) defines an anomaly as an observation that deviates markedly from the majority by a large distance in high-dimensional space. The algorithm starts by normalizing each time series to prevent variables with large variances having disproportional influence on Euclidean distances. The method uses the Leader algorithm (Hartigan, 1975) to identify anomalous clusters from which a representative member is selected. Nearest neighbour distances between the selected members are then calculated and form the primary source of information for the AD process. An extreme-value theory approach is used to calculate an anomalous threshold, which thus has a probabilistic interpretation.

The HDoultiers algorithm considers only the nearest neighbour distances to identify anomalies. Following Angiulli and Pizzuti (2002), Madsen (2018) proposed an algorithm using k nearest neighbour distances. For each observation, the k-nearest-neighbours (kNN) are first identified using a k-dimensional tree (kd-tree; Bentley, 1975) and an anomaly score is then calculated based on the distances to those neighbours. Whilst kNN-agg computes an aggregated distance to the kNN (see below), kNN-sum simply sums the distances to the kNN. The aggregated distance is calculated by aggregating the results from k-minimum-nearest neighbours ($k$minNN) to k-maximum nearest neighbours ($k$maxNN), such that if $k$minNN = 1 and $k$maxNN = 3, the results from 1NN, 2NN and 3NN are aggregated by taking the weighted sum, assigning nearest neighbours higher weights relative to the neighbours farther apart. Here, we used $k = 10$, the maximum default value of k in Madsen (2018) because it is a suitable trade-off between too low or high a value that may influence performance adversely (McCann and Lowe, 2012).

2.6. Select metrics to evaluate and compare methods (Step 6)

We selected several metrics to evaluate and compare the ability of the different AD methods outlined in Section 2.5, to detect the anomalies identified and labelled in Step 3 (Section 2.3), at the different sites for the different water-quality variables, anomaly classes and types (Table 2; Figs. S2–S3). We included commonly used metrics calculated easily from the confusion matrix of true and false positives and true and false negatives (TP, FP, TN, FP, respectively; Table S2). These included accuracy and error rate along with two metrics designed to better capture the performance of methods when the number of anomalous versus 'normal' observations is unbalanced, specifically the negative and positive predictive values (NPV and PPV, respectively; Ranawana and Palade, 2006). Finally, we used the root mean square error (RMSE) from the regression-based methods as an additional metric of performance for those methods.

Computation time can also provide insight on the comparative performance of methods. Both regression- and feature-based methods take time for classification. However, feature-based methods classify the complete time series in a single run. By contrast, regression-based methods require additional time for training for prediction, with the exception of naïve methods. Regression-based methods (barring naïve

prediction) also require additional time to perform optimization to estimate the model parameters; whilst this can be relatively fast for linear models, non-linear optimization is more time consuming. For these reasons, we can state a priori that running the feature-based methods will require less computational time than the regression-based methods. Furthermore, HDoutliers requires less computational time than both *k*NN methods because the former considers only the single most-nearest neighbour whereas the latter consider all *k* nearest neighbours. However, if the feature-based methods were to be implemented in near-real time to classify the time series with newly measured observations, this would make them more computationally comparable with regression-based methods, which are implemented in a loop that forecasts and classifies the one-step-ahead observation as anomalous or otherwise. As such, any difference in classification times between the approaches will depend on the models fitted and the features computed.

2.7. Prepare data for anomaly detection (Step 7)

Class 2 anomalies (i.e. impossible values of type F, out-of-sensor-range of type G and missing data of type K) were detected by the automated, hard-coded, classification rules in near-real time. For other anomalies, we implemented regression-based or feature-based methods. To prepare the 'clean' training data for the regression-based AD, we removed all the labelled anomalies from the time series data (Classes 1 and 3). Regression-based AD then followed using the natural log-transformed 'clean' time series for training and the natural log-transformed original times series for testing, for all methods except for linear autoregression for which we took the differences of the natural logarithms. These transformations were applied to meet assumptions of the regression models; forecasting was performed on the transformed scale. Where zero (e.g. type F anomalies in conductivity at PR) or negative values (e.g. type F anomalies in conductivity at PR and in level at SC) were present, we replaced each value with the (non-zero, positive) value of the previous observation to enable forecasting. To calculate the confusion-matrix based performance metrics for the regression-based methods, we first summed the 100% correctly detected Class 2 anomalies to the true positive (TP) count from the regression method before calculating the rest of the metrics (Table S2).

For feature-based AD, we applied both the one-sided and the two-sided derivative transformations to the natural log-transformed turbidity and conductivity time series because exploratory analyses indicated that these transformations highlighted the priority type A anomalies (e.g. large sudden spikes, Class 1) well in feature space. For the one-sided transformation, we took the negative side of the derivative for turbidity, and the positive side for conductivity. Feature-based AD on the transformed time series then followed. We followed the same process as for the regression-based methods, regarding the TP count, to calculate the complete set of confusion matrix-based performance metrics.

2.8. Implement anomaly detection methods (Step 8)

We used the *forecast* package (Hyndman, 2017) to implement the regression-based AD methods and the *DDoutliers* package (Madsen, 2018) run within the *oddwater* package (Talagala and Hyndman, 2018) to implement the feature-based AD methods in R statistical software (R Core Team, 2017). We used the same rule-based code to implement the automated classification rules within the regression- and feature-based methods. The R code for the automated classification rules and regression-based methods is provided in the Supplementary materials, along with files containing the time series data and anomaly-type coding. Madsen (2018) and Talagala and Hyndman (2018) describe the R code to implement the feature-based methods described herein.

3. Results

3.1. Anomalies and their types

Overall, we labelled 1651 turbidity, 521 conductivity and 8 level observations as anomalous in the time series data (Table 2). The majority of these anomalies were of type E (comprising periods of anomalous high variability), H (drift) and L (other).

There was imbalance in the number of non-anomalous (many) to anomalous (few) data points in the time series we used, as well as different types of anomalies (e.g. many type L vs few type A; Table 2). Furthermore, some anomaly types comprised multiple observations (e.g. other type L, drift type H) where as others contained only one (e.g. a type A anomaly). Such imbalances need to be considered in addition to the anomaly-type priority rankings when comparing and interpreting the performance of different methods with respect to their abilities to detect the different anomaly types.

The turbidity time series contained the most anomalies, at both SC and PR, followed by conductivity at PR. There were no clear examples of type C (constant offsets), although data labelled as L (other) between points of sudden shift may have been due to calibration errors manifesting as offsets. In addition, there were no examples of type G anomalies (out-of-sensor-range values). However, there were numerous impossible values (type F), which can be detected by automated classification rules in the same way as type G anomalies. Clusters of spikes (type I) and periods of low variability or persistent values (type B) were also absent. Type K anomalies (missing data) were present in all of the time series.

3.2. Evaluate and compare anomaly detection methods (Step 9)

We evaluated and compared results of the various AD methods as part of Step 9 of the AD Framework (Fig. 1), as outlined below.

3.2.1. Automated classification rules

As expected, the automated classification rules detected all of the Class 2 anomalies (types F, G and K; Table 2) correctly, with no false positives, in each of the time series.

3.2.2. Regression-based methods

Results of the regression-based methods performed on the turbidity and conductivity time series at both PR and SC indicated that, in general, all methods had high accuracy (values >0.80) and low error rates (<0.20), except when ADAM was used (Table 3). ADAM was associated with high rates of false positive detection (i.e. incorrect classification of normal observations as anomalies), which negatively affected the accuracy and error rates (Figs. 3–4 and S4–S9). For example, naïve prediction with ADAM applied to the turbidity time series at PR classified over 5000 observations as false positives compared to 133 without mitigation using AD alone (Table 3, Fig. 3). In many cases, large contiguous numbers of false positives occurred when the observations subsequent to a classified anomaly did not display 'normal' behaviour relative to the observations classified most recently as non-anomalous. Despite this drawback, ADAM was useful for correct classification of Class 3 anomalies where AD alone was not. For example, 718 out of 718 and 713 out of 915 type E (high variability) anomalies in the turbidity time series at PR and SC, respectively, were detected by naïve ADAM, and all 397 Type H (drift) and 80 type L (other) anomalies in the conductivity time series at PR were detected by ARIMA ADAM (Table 4). ADAM was also useful for detection of anomalous observations that proceeded sudden shifts, such as the L type anomalies in the middle of the turbidity time series (Figs. S2 and 3–4).

RegARIMA did not outperform ARIMA, despite the additional water-quality data that were used as covariates. This was especially true for conductivity at PR, where inclusion of other water-quality variables as covariates greatly reduced the rate of correct classification (RegARIMA

Table 3
Performance metrics for regression-based methods of anomaly detection performed separately on turbidity and conductivity data from in situ sensors at Pioneer River (PR) and Sandy Creek (SC), incorporating 100% detection of Class 2 anomalies by automated classification rules. See Tables S2–S3 for metric formulae and descriptions and Section 2.5.2 for model specifics. AD, anomaly detection; ADAM, anomaly detection and mitigation; AR, autoregression.

Site	Time series	Model (*p*,*d*,*q*)	Method	TN	FN	FP	TP	Accuracy	Error rate	NPV	PPV	RMSE
PR	Turbidity	Naïve (0,1,0)	AD	5416	715	133	16	0.86	0.14	0.88	0.11	0.21
			ADAM	347	0	5202	731	0.17	0.83	1.00	0.12	0.21
		Linear AR (4,0,0)	AD	5398	712	151	19	0.86	0.14	0.88	0.04	0.20
			ADAM	4491	25	1058	706	0.83	0.17	0.99	0.40	0.87
		ARIMA (3,1,2)	AD	5405	711	144	20	0.86	0.14	0.88	0.12	0.20
			ADAM	4465	25	1084	706	0.82	0.18	0.99	0.39	0.90
		RegARIMA (5,1,5)	AD	5344	695	205	36	0.86	0.14	0.88	0.15	0.57
			ADAM	171	0	5378	731	0.14	0.86	1.00	0.12	0.39
PR	Conductivity	Naïve (0,1,0)	AD	5759	459	2	60	0.93	0.07	0.93	0.97	0.17
			ADAM	4455	399	1306	120	0.73	0.27	0.92	0.08	0.17
		Linear AR (2,0,0)	AD	5709	453	52	66	0.92	0.08	0.93	0.56	0.17
			ADAM	4256	397	1505	122	0.70	0.30	0.91	0.07	0.64
		ARIMA(3,1,2)	AD	5756	455	5	64	0.93	0.07	0.93	0.93	0.16
			ADAM	1873	0	3888	519	0.38	0.62	1.00	0.12	1.37
		RegARIMA (1,1,2)	AD	5675	437	86	82	0.92	0.08	0.93	0.49	0.26
			ADAM	128	0	5633	519	0.10	0.90	1.00	0.08	0.07
SC	Turbidity	Naïve (0,1,0)	AD	4386	859	96	61	0.82	0.18	0.84	0.39	0.24
			ADAM	491	134	3991	786	0.24	0.76	0.79	0.16	0.24
		Linear AR (5,0,0)	AD	4347	830	135	90	0.82	0.18	0.84	0.40	0.22
			ADAM	2178	753	2340	167	0.43	0.57	0.74	0.07	1.06
		ARIMA (3,1,2)	AD	4348	829	134	91	0.82	0.18	0.84	0.40	0.22
			ADAM	2187	751	2295	169	0.44	0.56	0.74	0.07	1.06
		RegARIMA (5,1,0)	AD	4345	820	137	100	0.82	0.18	0.84	0.42	0.23
			ADAM	775	81	3707	839	0.30	0.70	0.91	0.18	0.06

PPV of 0.49 vs ARIMA PPV of 0.93; Table 3). This likely reflects the characteristics of the water-quality time series at this site, with conductivity displaying complex relationships with both turbidity and level (Fig. S1). Thus, including these covariates had a detrimental impact on classification performance. In addition, the behaviour of conductivity tended to be more stable than turbidity through time, somewhat reflective of random walk behaviour, on which naïve prediction (ARIMA(0,1,0)) is based (Hyndman and Koehler, 2006). This may be why the ARIMA (3,1,2) model performed similarly well to naïve prediction when applied to the conductivity time series at PR, given both were using a difference (*d*) parameter of 1 (Table 3, Figs. 3–4).

There were only two observations labelled as anomalies in the conductivity time series at SC, and both were of Class 1 (one sudden spike A and one sudden shift D). These two anomalies were classified correctly by all methods, with zero false negatives (Table 4, Table S4). However, all of the methods classified many 'normal' observations incorrectly as anomalies (false positives), particularly ADAM (up to 5091 out of 6280 observations; Table S4), as was the case for other time series at

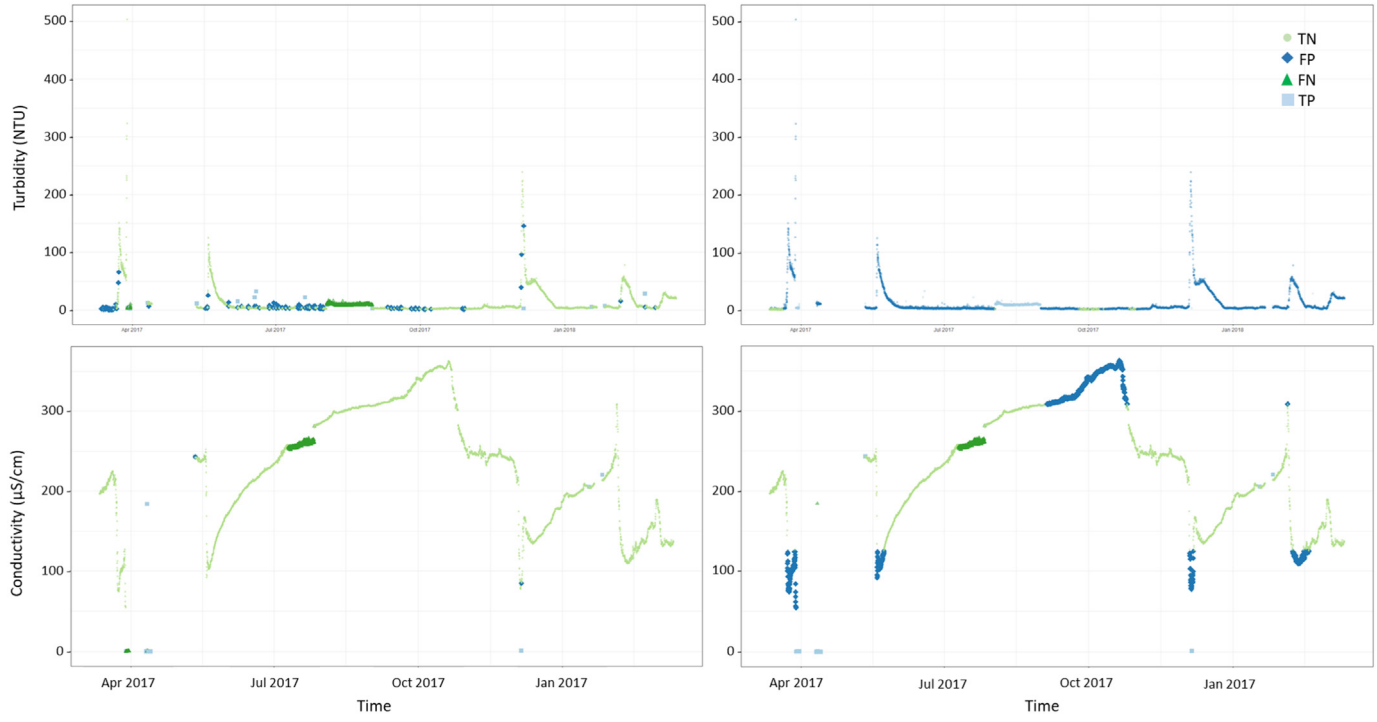


Fig. 3. Classification of turbidity (upper row) and conductivity observations (lower row) measured by in situ sensors at Pioneer River (PR) by naïve prediction as true negatives (TN), false negatives (FN), false positives (FP) or true positives (TP). Plots on the left show results from naïve prediction alone, those on the right show results from naïve prediction with ADAM.

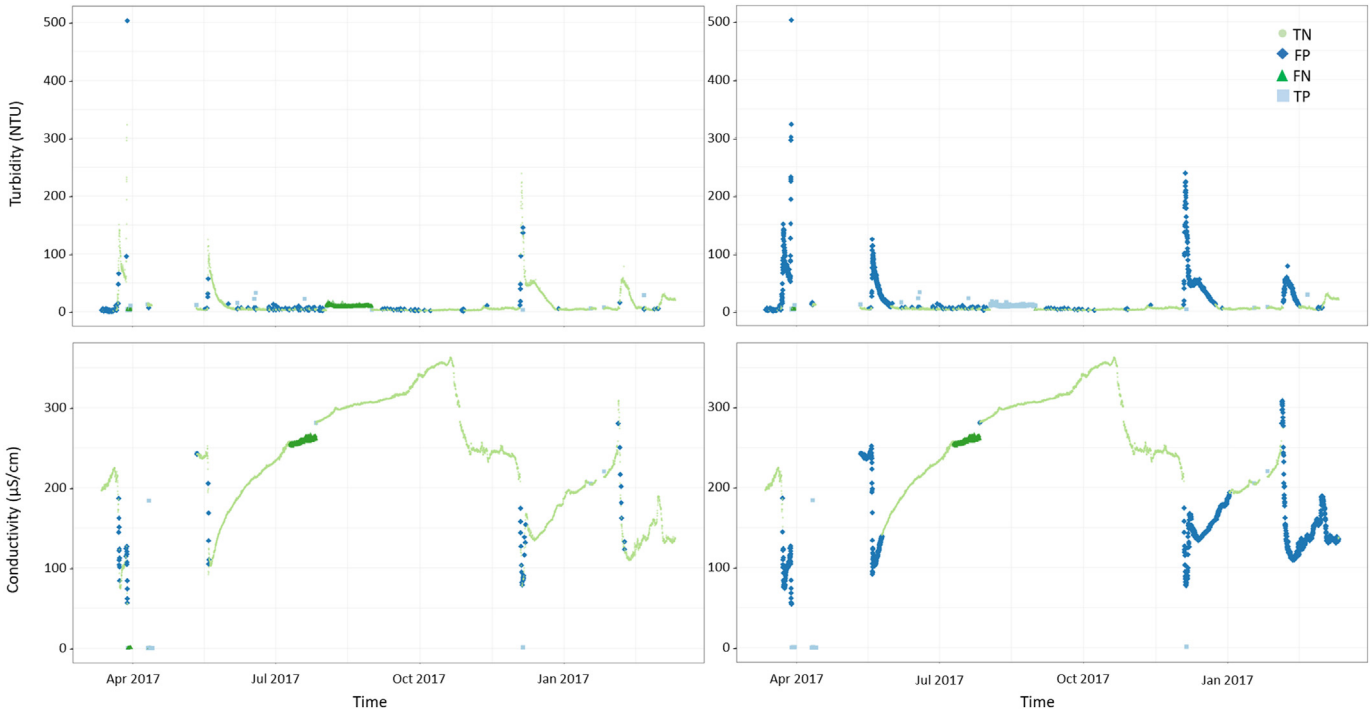


Fig. 4. Classification of turbidity (upper row) and conductivity observations (lower row) measured by in situ sensors at Pioneer River (PR) by ARIMA as true negatives (TN), false negatives (FN), false positives (FP) or true positives (TP). Plots on the left show results from ARIMA alone, those on the right show results from ARIMA with ADAM.

Table 4
Number of turbidity (T) and conductivity (C) anomalies of each type and class classified correctly by each regression-based method for Pioneer River (PR) and Sandy Creek (SC). Number of true anomalies and number of instances where relevant indicated in parentheses. Class 2 anomalies detected by automated classification rules. AR, autoregression, — not applicable.

River (variable)	Model	Method	A Class 1	D Class 1	E Class 3	F Class 2	J Class 1	K Class 2	H Class 3	L Class 3
PR (T)			(1)	(3)	(0)	(0)	(5)	(4)	(0)	(718; 1 instance)
	Naïve	AD	1	3	–	–	5	4	–	3
		ADAM	1	3	–	–	5	4	–	718
	Linear AR	AD	1	3	–	–	5	4	–	6
		ADAM	1	2	–	–	5	4	–	694
	ARIMA	AD	1	3	–	–	5	4	–	7
		ADAM	1	3	–	–	5	4	–	694
	RegARIMA	AD	1	3	–	–	5	4	–	23
		ADAM	1	3	–	–	5	4	–	718
PR (C)			(2)	(2)	(0)	(34)	(0)	(4)	(397; 1 instance)	(80; 2 instances)
	Naïve	AD	2	1	–	34	–	4	0	19
		ADAM	1	1	–	34	–	4	0	80
	Linear AR	AD	2	2	–	34	–	4	0	24
		ADAM	2	2	–	34	–	4	0	80
	ARIMA	AD	2	1	–	34	–	4	0	23
		ADAM	2	2	–	34	–	4	397	80
	RegARIMA	AD	2	2	–	34	–	4	0	40
		ADAM	2	2	–	34	–	4	397	80
SC (T)			(4)	(1)	(915; 2 instances)	(0)	(0)	(0)	(0)	(0)
	Naïve	AD	4	1	276	–	–	–	–	–
		ADAM	4	1	780	–	–	–	–	–
	Linear AR	AD	4	0	85	–	–	–	–	–
		ADAM	4	1	161	–	–	–	–	–
	ARIMA	AD	4	0	85	–	–	–	–	–
		ADAM	4	1	162	–	–	–	–	–
	RegARIMA	AD	4	1	94	–	–	–	–	–
		ADAM	4	1	833	–	–	–	–	–
SC (C)			(1)	(1)	(0)	(0)	(0)	(0)	(0)	(0)
	Naïve	AD	1	1	–	–	–	–	–	–
		ADAM	1	1	–	–	–	–	–	–
	Linear AR	AD	1	1	–	–	–	–	–	–
		ADAM	1	1	–	–	–	–	–	–
	ARIMA	AD	1	1	–	–	–	–	–	–
		ADAM	1	1	–	–	–	–	–	–
	RegARIMA	AD	1	1	–	–	–	–	–	–
		ADAM	1	1	–	–	–	–	–	–

Table 5
Performance metrics for feature-based methods of anomaly detection performed on multivariate water-quality time series from in situ sensors at Pioneer River (PR) and Sandy Creek (SC), incorporating 100% detection of Class 2 anomalies by automated classification rules. See Tables S2–S3 for metric formulae and descriptions. OS, one sided.

Site	Time series	Method	Transformation	TN	FN	FP	TP	Accuracy	Error rate	NPV	PPV
PR	Turbidity	HDoutliers	Derivative	5548	728	1	3	0.88	0.12	0.75	0.88
	Turbidity		OS Derivative	5547	727	2	4	0.88	0.12	0.67	0.88
	Turbidity	*k*NN-agg	Derivative	5542	725	7	6	0.88	0.12	0.46	0.88
	Turbidity		OS Derivative	5546	728	3	3	0.88	0.12	0.50	0.88
	Turbidity	*k*NN-sum	Derivative	5547	728	2	3	0.88	0.12	0.60	0.88
	Turbidity		OS Derivative	5546	728	3	3	0.88	0.12	0.50	0.88
	Conductivity	HDoutliers	Derivative	5758	470	3	49	0.92	0.08	0.94	0.92
	Conductivity		OS Derivative	5758	479	3	40	0.92	0.08	0.93	0.92
	Conductivity	*k*NN-agg	Derivative	5759	472	2	47	0.92	0.08	0.96	0.92
	Conductivity		OS Derivative	5758	479	3	40	0.92	0.08	0.93	0.92
	Conductivity	*k*NN-sum	Derivative	5760	471	1	48	0.92	0.08	0.98	0.92
	Conductivity		OS Derivative	5759	479	2	40	0.92	0.08	0.95	0.92
SC	Turbidity	HDoutliers	Derivative	4477	914	5	6	0.83	0.17	0.55	0.83
	Turbidity		OS Derivative	4481	917	1	3	0.83	0.17	0.75	0.83
	Turbidity	*k*NN-agg	Derivative	4477	914	5	6	0.83	0.17	0.55	0.83
	Turbidity		OS Derivative	4471	912	11	8	0.83	0.17	0.42	0.83
	Turbidity	*k*NN-sum	Derivative	4482	920	0	0	0.83	0.17	n/a	0.83
	Turbidity		OS Derivative	4480	917	2	3	0.83	0.17	0.60	0.83

both SC and PR (Table 3). Due to the heavy imbalance of normal versus anomalous observations in the conductivity data at SC, we decided not to undertake further interpretation of the regression-based performance metrics for this time series.

Diagnostics conducted on the residuals of each regression-based method (Figs. S13–S20) indicated heteroscedasticity was present. In other words, there was change in variance of the data through time (a form of nonstationarity), despite the transformations applied to the time series. Although this will not bias the model forecasts, it may have reduced the accuracy of the prediction intervals and hence affected the classification of anomalies. There was also evidence of nonstationarity in terms of non-constant means in the PR turbidity and conductivity residuals from the linear autorgression, ARIMA and RegARIMA and in the SC turbidity residuals from the ARIMA and RegARIMA models (Box-Ljung tests, $p < 0.05$).

3.2.3. Feature-based methods

Each feature-based method applied to the turbidity time series at PR had the same accuracy (0.88), error rate (0.12) and NPV score (0.88; Table 5; Fig. 5). The *k*NN-agg method applied to the derivatives of the time series correctly classified the most anomalies (6) of all feature-based methods applied to the PR turbidity data, but also resulted in the most false positives (7) and thus the lowest NPV score (0.46). The HDoutliers method applied to the derivatives of the time series attained the highest NPV score of 0.75, thus attaining the highest values of both NPV and PPV. All methods had high rates of false negative detection

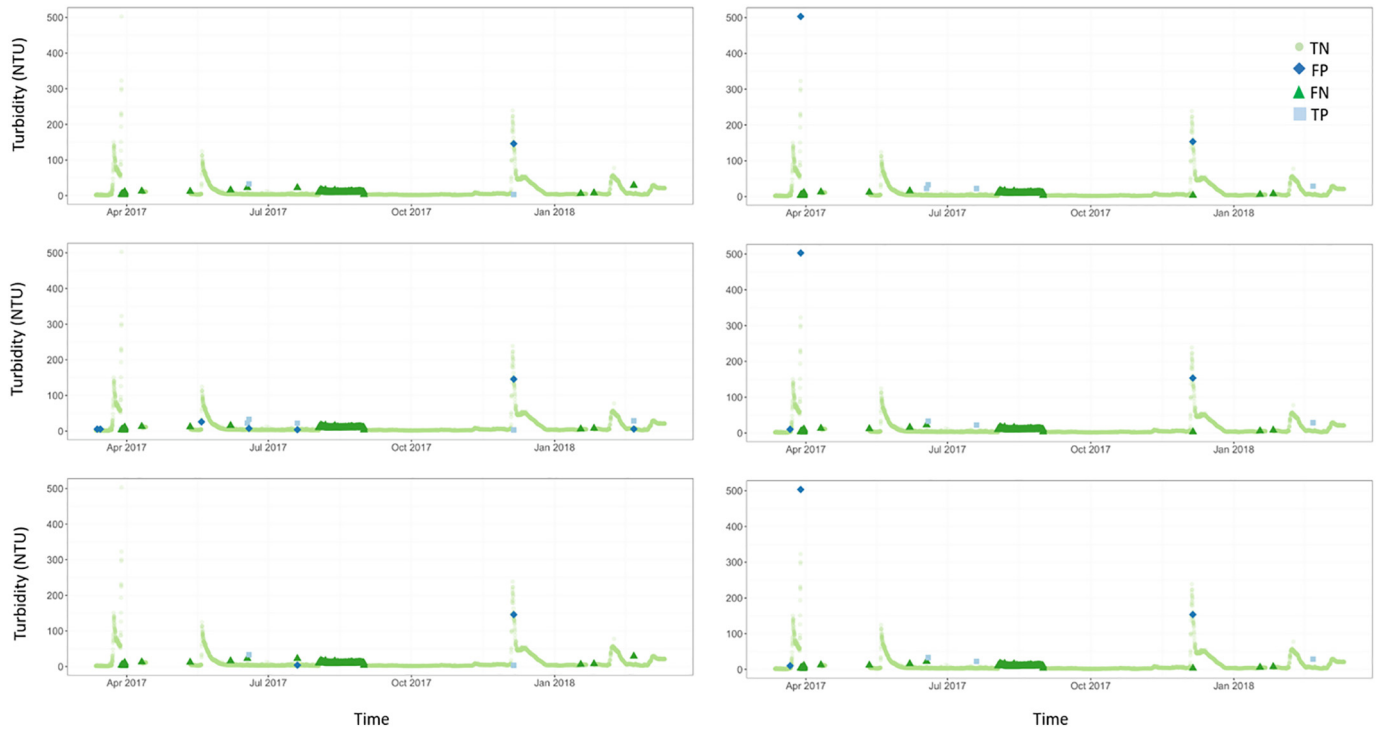


Fig. 5. Classification of turbidity measured by an in situ sensor at Pioneer River (PR) by HDoutliers (upper row), *k*NN-agg (middle row) and *k*NN-sum (lower row) as true negatives (TN), false negatives (FN), false positives (FP) or true positives (TP). Plots on the left show results of methods applied to the derivatives, and those on the right show results of methods applied to the one-sided derivatives of the time series.

Table 6
Number of turbidity (T) and conductivity (C) anomalies of each type and class classified correctly by each feature-based method for Pioneer River (PR) and Sandy Creek (SC). Number of Pioneer River (PR) turbidity anomalies of each type and class classified correctly by each feature-based method. Number of true anomalies and number of instances where relevant indicated in parentheses. Class 2 anomalies detected by automated classification rules. –, not applicable.

River (variable)	Method	Transformation	A Class 1	D Class 1	E Class 3	F Class 2	J Class 1	K Class 2	H Class 3	L Class 3
PR (T)			(1)	(3)	(0)	(0)	(5)	(4)	(0)	(718; 1 instance)
	HDoutliers	Derivative	1	1	–	–	1	4	–	0
		OS Derivative	0	0	–	–	4	4	–	0
	*k*NN-agg	Derivative	1	1	–	–	4	4	–	0
		OS Derivative	0	0	–	–	3	4	–	0
	*k*NN-sum	Derivative	1	1	–	–	1	4	–	0
		OS Derivative	0	0	–	–	3	4	–	0
PR (C)			(2)	(2)	(0)	(34)	(0)	(4)	(397; 1 instance)	(80; 2 instances)
	HDoutliers	Derivative	1	1	–	34	–	4	0	12
		OS Derivative	1	0	–	34	–	4	0	4
	*k*NN-agg	Derivative	1	1	–	34	–	4	0	10
		OS Derivative	1	1	–	34	–	4	0	4
	*k*NN-sum	Derivative	1	1	–	34	–	4	0	11
		OS Derivative	0	0	–	34	–	4	0	0
SC (T)			(4)	(1)	(915; 2 instances)	(0)	(0)	(0)	(0)	(0)
	HDoutliers	Derivative	3	0	3	–	–	–	–	–
		OS Derivative	3	0	0	–	–	–	–	–
	*k*NN-agg	Derivative	3	0	3	–	–	–	–	–
		OS Derivative	3	0	5	–	–	–	–	–
	*k*NN-sum	Derivative	0	0	0	–	–	–	–	–
		OS Derivative	3	0	0	–	–	–	–	–
SC (C)			(1)	(1)	(0)	(0)	(0)	(0)	(0)	(0)
	HDoutliers	Derivative	1	0	–	–	–	–	–	–
		OS Derivative	1	0	–	–	–	–	–	–
	*k*NN-agg	Derivative	1	0	–	–	–	–	–	–
		OS Derivative	1	0	–	–	–	–	–	–
	*k*NN-sum	Derivative	1	0	–	–	–	–	–	–
		OS Derivative	1	0	–	–	–	–	–	–

(>720; Table 5), which were associated predominantly with poor detection of Class 3 anomalies; none of the 718 type L ('other') anomalies were classified as such in the turbidity times series by any feature-based method (Table 6, Fig. 4). Furthermore, only the methods applied to the derivatives of the turbidity time series correctly classified the type A (sudden spike) and one of the type D (sudden shift) anomalies (Table 6).

For conductivity at PR, accuracy was high (0.92) and error rate was low (0.08; Table 5; Fig. S10). The PPV values were all identical and high (0.92), with slightly more variability in the NPV scores, which were also high (0.93–0.98); the *k*NN-sum method on the one-sided derivatives attained the highest NPV. However, the feature-based methods tended to produce high false negative rates for the conductivity data, as was the case with the turbidity data at PR. Most methods were able to correctly classify the type A and D anomalies (Table 6).

For turbidity at SC, accuracy (0.83) and error rate (0.17) were the same for each method, as was the case for turbidity at PR (Table 5; Fig. S11). NPV scores ranged from 0.42, attained by the *k*NN-agg method, to 0.75 attained by HDoutliers, both of which were applied to the one-sided derivatives of the time series (Table 5). All methods had high false negative rates (>900; Table 5), but all methods classified three of the four type A anomalies correctly (Table 6).

For the feature-based methods applied to the conductivity time series at SC, we followed the same protocol as we did for the regression-based methods (Table S5, Fig. S12), keeping interpretation to a minimum given there were only two anomalies labelled in these data. All methods classified one true positive only (Type A) and misclassified the other anomaly (type D) as normal, but most non-anomalous observations were classified correctly as true negatives (Tables 6 and S5).

4. Discussion

The tenth and final step of the AD Framework (Fig. 1) involves making recommendations based on the results of the different AD methods applied. Here, results of the regression-based methods indicated that the ARIMA method may be useful for AD in streaming water-quality data because it encompasses both naïve prediction (ARIMA(*0,1,0*)) and linear autoregression models (ARIMA(*p,0,0*)) within its suite of possible models. Furthermore, ARIMA may be particularly useful when no other covariates are available to include in RegARIMA models, relationships among potential covariates are complex, such as at PR, or covariates contain missing values. Regarding decisions on whether to include anomaly mitigation as well as detection, ARIMA without mitigation (i.e. without ADAM) may be most useful when the end-user focus is on detection of Class 1 anomalies (sudden spikes and shifts). Such anomalies, if not detected and accounted for, are likely to incorrectly inflate or deflate summary statistics (e.g. monthly means) used in water quality assessments and for compliance checking by water management agencies.

ARIMA with mitigation (i.e. with ADAM) could be implemented subsequently or alternatively to ARIMA alone to detect Class 3 anomalies (e.g. drifts, periods of high variability). Occurrence of such anomalies can indicate that sensors need re-calibrating, and their detection would be of particular value in terms of sensor maintenance. ARIMA models assume that observations are evenly spaced in time, which may become problematic for the models, specifically for the characteristics of the training datasets, if in situ water-quality measurements become less frequent and/or irregular in time. This may be especially problematic in training datasets if natural water-quality events are missed. However, increasing the frequency of measurements during high-flow events to capture greater resolution in water-quality dynamics is less problematic. Most importantly, the training dataset should include the full range of natural water-quality dynamics.

Regression-based methods of AD are semi-supervised, and as a result are influenced strongly by the training data used to build the models. In this case study, high rates of false positives were detected in the water-quality time series when these methods of AD were used (Table S1). Yet, decisions on how to dissect time series data into training and test components are not trivial, particularly when there are time-specific behaviours in the data such as seasonality of events and/or

when the time series are of limited length (e.g. one year, as was the case here). Methods such as event-based cross validation (Lessels and Bishop, 2013) and walk-forward cross-validation (Bergmeir et al., 2018) may provide potential solutions that could be implemented in future research.

In our analysis, the regression models may have been over-fitted because they were trained on the same data (minus anomalies) used for testing. Using training data from another nearby site on the same watercourse or from a different time period at the same site could lessen this issue. However, there were no nearby sites on PR or SC from which water-quality data from in situ sensors were available. If such data become available in the future, training could be performed on those data to see if the AD performance changes.

In rivers, water-quality patterns through time often change with the flow regime (Poff et al., 1997; Nilsson and Renöfält, 2008). This is particularly apparent in highly seasonal rivers such as those of Australia's tropical north, where water quality tends to fluctuate more rapidly and to a greater extent during high-flow events in the wet season than during the more stable low-flow phase of the dry season (Leigh, 2013; O'Brien et al., 2016). This can manifest as nonstationarity in the water-quality time series; for example, as changing variance through time, as was the case here. As such, differentiating between regimes for training purposes may additionally improve the performance of regression-based AD methods in water-quality time series from such rivers. For example, discrete-space hidden Markov models could be used to classify (i.e. segment) the time series into a subset of water-quality regimes found in the data. The regression-based models that require a training dataset (Table S1) could then be applied subsequently to each of the segmented datasets.

Like the regression-based methods without ADAM, the feature-based methods we implemented were not proficient at detecting Class 3 anomalies. This is not surprising given the transformations and algorithms used to implement these methods were developed specifically to prioritize detection of Class 1 anomalies, as per the end-user needs and goals in our case study. Other transformations of the time series data may be required to better target Class 3 anomalies using feature-based methods. This should be borne in mind when transferring our approach to other applications and end-user objectives, such as the monitoring and detection of security intrusions (García-Teodoro et al., 2009; Talagala et al., 2018). Furthermore, whilst HDoutliers is more computationally efficient than *k*NN methods of feature-based AD, *k*NN methods may be preferable when clusters of anomalies are present in the high-dimensional feature-space produced from the transformed time-series data. Such clusters could manifest if, for example, there were several sudden spikes in the time series, each of the same value. Such phenomena may result from recurrent technical issues with the sensor equipment that produce a specific, recurrent anomalous value.

5. Conclusions

Our results highlight that a combination of methods, as recommended in Section 4, is likely to provide optimal performance in terms of correct classification of anomalies in streaming water-quality data from in situ sensors, whilst minimizing false detection rates. Furthermore, our framework emphasizes the importance of communication between end-users and anomaly detection developers for optimal outcomes with respect to both detection performance and end-user application. To this end, our framework has high transferability to other types of high frequency time-series data and anomaly detection applications. Within the purview of water-quality monitoring, for example, our framework could be applied to other water-quality variables measured by in situ sensors that are used commonly in ecosystem health assessments, such as dissolved oxygen, water temperature and nitrate (Leigh et al., 2013; Pellerin et al., 2016). These properties of water are highly dynamic in space and time (Hunter and Walton, 2008; Boulton et al., 2014) and so differentiating anomalies from real water-quality events may be more challenging than it is for properties like turbidity and conductivity investigated in this study. These latter two properties hold promise as near-real time surrogates of sediment and nutrient concentrations (Jones et al., 2011; Slaets et al., 2014), which would reduce the amount of laboratory analysis otherwise required for discrete water samples. Therefore, the extension of automated AD methods, as developed herein, along with models that predict sediment and nutrient concentrations from these data, into space and time on river networks, could revolutionize the way we monitor and manage water quality, whilst also increasing scientific understanding of the spatio-temporal dynamics of water-quality in rivers and the potential effects on downstream waters.

Acknowledgements

Funding for this project was provided by the Queensland Department of Environment and Science (DES) and the ARC Centre of Excellence for Mathematical and Statistical Frontiers (ACEMS). A repository of the water-quality data from the in situ sensors used herein and the code used to implement the regression-based anomaly detection methods are provided in the Supplementary materials.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.scitotenv.2019.02.085.

References

Akaike, H., 1974. A new look at the statistical model identification. IEEE Trans. Autom. Control 19, 716–723.
Angiulli, F., Pizzuti, C., 2002. Fast outlier detection in high dimensional spaces. European Conference on Principles of Data Mining and Knowledge Discovery. Springer, Berlin, Heidelberg, pp. 15–27.
Bentley, J.L., 1975. Multidimensional binary search trees used for associative searching. Commun. ACM 18, 509–517.
Bergmeir, C., Hyndman, R.J., Koo, B., 2018. A note on the validity of cross-validation for evaluating autoregressive time series prediction. Comput. Stat. Data Anal. 120, 70–83.
Boulton, A., Brock, M., Robson, B., Ryder, D., Chambers, J., Davis, J., 2014. Australian Freshwater Ecology: Processes and Management. John Wiley & Sons.
Box, G.E.P., Jenkins, G.M., 1970. Time Series Analysis: Forecasting and Control. Holden-Day Incorporated, San Francisco, CA, USA.
Brodie, J., 2004. Mackay Whitsunday region: state of the waterways. ACTFR Technical Report No. 02/03. Australian Centre for Tropical Freshwater Research, James Cook University, Townsville, Australia.
Chandola, V., Banerjee, A., Kumar, V., 2009. Anomaly detection: a survey. ACM Computing Surveys (CSUR). vol. 41, p. 15.
Cox, B.A., 2003. A review of currently available in-stream water-quality models and their applicability for simulating dissolved oxygen in lowland rivers. Sci. Total Environ. 314, 335–377.
Fiebrich, C.A., Morgan, C.R., McCombs, A.G., Hall Jr., P.K., McPherson, R.A., 2010. Quality assurance procedures for mesoscale meteorological data. J. Atmos. Ocean. Technol. 27, 1565–1582.
Gandomi, A., Haider, M., 2015. Beyond the hype: big data concepts, methods, and analytics. Int. J. Inf. Manag. 35, 137–144.
García-Teodoro, P., Diaz-Verdejo, J., Maciá-Fernández, G., Vázquez, E., 2009. Anomaly-based network intrusion detection: techniques, systems and challenges. Comput. Secur. 28, 18–28.
Goldstein, M., Uchida, S., 2016. A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. PLoS One 11, e0152173.
Hartigan, J.A., 1975. Clustering Algorithms. Wiley, New York, NY, USA.
Hill, D.J., Minsker, B.S., 2010. Anomaly detection in streaming environmental sensor data: a data-driven modeling approach. Environ. Model. Softw. 25, 1014–1022.
Hill, D.J., Minsker, B.S., Amir, E., 2009. Real-time Bayesian anomaly detection in streaming environmental data. Water Resour. Res. 45, W00D28.
Horsburgh, J.S., Reeder, S.L., Jones, A.S., Meline, J., 2015. Open source software for visualization and quality control of continuous hydrologic and water quality sensor data. Environ. Model. Softw. 70, 32–44.
Hunter, H.M., Walton, R.S., 2008. Land-use effects on fluxes of suspended sediment, nitrogen and phosphorus from a river catchment of the Great Barrier Reef, Australia. J. Hydrol. 356, 131–146.
Hyndman, R.J., 2017. forecast: forecasting functions for time series and linear models. R package version 8.2. http://pkg.robjhyndman.com/forecast.
Hyndman, R.J., Athanasopoulos, G., 2018. Forecasting: Principles and Practice. 2nd edition. OTexts https://OTexts.org/fpp2/.
Hyndman, R.J., Koehler, A.B., 2006. Another look at measures of forecast accuracy. Int. J. Forecast. 22, 679–688.

Jones, A.S., Stevens, D.K., Horsburgh, J.S., Mesner, N.O., 2011. Surrogate measures for providing high frequency estimates of total suspended solids and total phosphorus concentrations. J. Am. Water Resour. Assoc. 47, 239–253.
Leigh, C., 2013. Dry-season changes in macroinvertebrate assemblages of highly seasonal rivers: responses to low flow, no flow and antecedent hydrology. Hydrobiologia 703, 95–112.
Leigh, C., Burford, M.A., Connolly, R.M., Olley, J.M., Saeck, E., Sheldon, F., Smart, J.C.R., Bunn, S.E., 2013. Science to support management of receiving waters in an event-driven ecosystem: from land to river to sea. Water 5, 780–797.
Lessels, J.S., Bishop, T.F.A., 2013. Estimating water quality using linear mixed models with stream discharge and turbidity. J. Hydrol. 498, 13–22.
Li, J., Pedrycz, W., Jamal, I., 2017. Multivariate time series anomaly detection: a framework of Hidden Markov Models. Appl. Soft Comput. 60, 229–240.
Liu, S., McGree, J.M., Ge, Z., Xie, Y., 2015. Computational and Statistical Methods for Analysing Big Data With Applications. Academic Press, London.
Ljung, G., Box, G., 1978. On a measure of lack of fit in time series models. Biometrika 65, 297–303.
Madsen, J.H., 2018. DDoutlier: distance & density-based outlier detection. R package version 0.1.0. https://CRAN.R-project.org/package=DDoutlier.
McCann, S., Lowe, D.G., 2012. Local naive Bayes nearest neighbor for image classification. IEEE Conference on Computer Vision and Pattern Recognition. IEEE, Vancouver, BC, Canada, pp. 3650–3656.
Moatar, F., Miquel, J., Poirel, A., 2001. A quality-control method for physical and chemical monitoring data. Application to dissolved oxygen levels in the river Loire (France). J. Hydrol. 252, 25–36.
Nilsson, C., Renöfält, B.M., 2008. Linking flow regime and water quality in rivers: a challenge to adaptive catchment management. Ecol. Soc. 13, 18.
O'Brien, K.R., Weber, T.R., Leigh, C., Burford, M.A., 2016. Sediment and nutrient budgets are inherently dynamic: evidence from a long-term study of two subtropical reservoirs. Hydrol. Earth Syst. Sci. 20, 4881–4894.
Pellerin, B.A., Stauffer, B.A., Young, D.A., Sullivan, D.J., Bricker, S.B., Walbridge, M.R., Clyde Jr., G.A., Shaw, D.M., 2016. Emerging tools for continuous nutrient monitoring networks: sensors advancing science and water resources protection. J. Am. Water Resour. Assoc. 52, 993–1008.
Poff, N.L., Allan, J.D., Bain, M.B., Karr, J.R., Prestegaard, K.L., Richter, B.D., Sparks, R.E., Stromberg, J.C., 1997. The natural flow regime. Bioscience 47, 769–784.
R Core Team, 2017. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria https://www.R-project.org/.
Rabinovici, S.J.M., Bernknopf, R.L., Wein, A.M., Coursey, D.L., Whitman, R.L., 2004. Economic and health risk trade-offs of swim closures at a Lake Michigan beach. Environ. Sci. Technol. 38, 2737–2745.
Ranawana, R., Palade, V., 2006. Optimized precision: a new measure for classifier performance evaluation. IEEE Congress on Evolutionary Computation. IEEE, Vancouver, BC, Canada, pp. 2254–2261.
Slaets, J.I., Schmitter, P., Hilger, T., Lamers, M., Piepho, H.P., Vien, T.D., Cadisch, G., 2014. A turbidity-based method to continuously monitor sediment, carbon and nitrogen flows in mountainous watersheds. J. Hydrol. 513, 45–57.
Talagala, P.D., Hyndman, R.J., 2018. oddwater: a package for outlier detection in water quality sensor data. GitHub Repository https://github.com/pridiltal/oddwater.
Talagala, P., Hyndman, R., Smith-Miles, K., Kandanaarachchi, S., Munoz, M., 2018. Anomaly Detection in Streaming Nonstationary Temporal Data. Working Paper No. 4/18. Department of Econometrics and Business Statistics, Monash University.
Tsay, R.S., 1988. Outliers, level shifts, and variance changes in time series. J. Forecast. 7, 1–20.
Tsay, R.S., 1989. Testing and modeling threshold autoregressive processes. J. Am. Stat. Assoc. 84, 231–240.
Wilkinson, L., 2018. Visualizing big data outliers through distributed aggregation. IEEE Trans. Vis. Comput. Graph. 24, 256–266.

Chapter 6

Conclusion

This thesis by publication is built around four articles. Although the four articles have their own focus motivated by a wide range of different analytical challenges from different fields, none of them is completely an anomaly. The four articles move around a unifying theme on anomaly detection in streaming time series data, with a different degree of attention to the common theme.

6.1 Summary of the Results and Contributions

Despite the long history of research on anomaly detection, the problem is still challenging owing to the evolving nature of the problem setting introduced by different applications and user requirements. This thesis is an attempt to reduce this gap by introducing three new algorithms, stray, oddstream and oddwater, for anomaly detection in temporal data with applications in pedestrian monitoring, security monitoring and sensor quality monitoring, respectively. The three algorithms stem from the analytical challenges introduced by the applications with various input data structures, definitions, problem specifications, user requirements, limitations of the state-of-the-art methods and unavailability of techniques that accommodate some of the data challenges.

6.1.1 The stray algorithm

Anomaly detection in high-dimensional data is a challenging yet important task, because it has applications in many fields. The HDoutliers algorithm by Wilkinson (2018) is a powerful algorithm for anomaly detection in high-dimensional data with a strong theoretical foundation. However, it suffers from a few limitations since it limits the anomalous score calculation only to the nearest neighbour distances and uses the Leader algorithm to form several clusters of points prior to anomalous score calculation. The effect of these limitations is a tendency to reduce computational efficiency and increase false detection rates under certain circumstances. Therefore, the main objective of Chapter 2 was to propose solutions to the limitation of the HDoutliers algorithm and thereby improve its capabilities.

The proposed algorithm, stray, addresses the limitations of the HDoutliers algorithm. In the stray algorithm, an anomaly is defined as an observation that deviates markedly from the majority with a large distance gap. It calculates an anomalous score for each data instance using k-nearest neighbour distances with the maximum gap. An approach based on extreme value theory is then applied to the anomalous scores to calculate a data-driven anomalous threshold. This improved algorithm can assign both a label and an anomalous score that explains the level of outlierness of each data instance.

This study offers two fundamental contributions. First, it proposes an improved algorithm for anomaly detection in high-dimensional data that addresses the limitations of the state-of-art-method, the HDoutliers algorithm. It outperforms the state-of-the-art method in both accuracy and computational efficiency. Among many other advantages, the stray algorithm has the ability to deal with the masking problem, multimodal distributions and inliers and outliers. The stray algorithm is specially designed for high-dimensional data. As the second contribution, the study demonstrates how the stray algorithm can assist in detecting anomalies present in other data structures, such as temporal data and streaming data, using feature engineering.

Since the stray algorithm is based on the distance definition of an anomaly, the algorithm expects data instances to have a clear distance separation between the anomalous and

typical points. Then, only the anomalous points (if any) have significantly large k-nearest neighbour distances with the maximum gap that discriminate anomalies from typical points. However, some applications do not exhibit large gaps between typical points 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 points. In the absence of clear distance separation between anomalous points and the typical points, the stray algorithm fails to detect anomalies since distance measures are the primary source of information for the algorithm to detect anomalies. This limitation of the stray algorithm motivates the second algorithm proposed in Chapter 3 of this thesis.

6.1.2 The oddstream algorithm

In addition to the aforementioned limitation of the stray algorithm, the limited research attempts for detecting anomalous series within a large collection of streaming data motivated the second algorithm of this thesis, the oddstream algorithm. The primary focus of Chapter 3 was to develop a powerful automated method to detect anomalous series within a large collection of series in the streaming data context.

In the oddstream algorithm, an anomaly is defined as an observation that is very unlikely, given the recent distribution of a given system. In this algorithm, a boundary for the system's typical behaviour is defined using extreme value theory. Then, a sliding window is used to test newly arrived data. The model uses time series features as inputs and a density-based comparison to locate nonstationarity. This algorithm can detect significant changes in the typical behaviour and automatically update the anomalous threshold on detecting nonstationarity.

This study offers three fundamental contributions. First, it proposes a new framework that provides early detection of anomalies within a large collection of streaming time series data. Second, it proposes a novel approach that adapts to nonstationarity. Third, using various synthetic and real datasets, it demonstrates the wide applicability and usefulness of the algorithm. Application of the oddstream algorithm with data obtained using fibre optic cables for intrusion detection showed that the algorithm has the ability to deal with large nonstationary streaming data that may have multimodal distributions.

6.1.3 The oddwater algorithm

Automated *in situ* sensors have the potential to revolutionise the way we monitor environmental conditions. However, the data produced by these sensors are prone to errors because of many reasons, such as miscalibration, biofouling and battery failures (Horsburgh et al., 2015). These technical outliers make the data unreliable for scientific analysis. Therefore, to ensure water-quality sensors yield high-quality data, we need to automate the real-time detection of technical outliers in such data. However, a customised method to detect technical outliers in water-quality data from *in situ* sensors is lacking. No exiting outlier detection method is able to address this challenge owing to the complex nature of the definition of a technical outlier in water-quality data from *in situ* sensors. Therefore, the main objective of Chapters 4 and 5 was to propose a new framework to detect technical outliers in high-frequency water-quality data from *in situ* sensors.

This study proposes an automated framework that provides early detection of technical outliers, caused by technical issues, in water-quality data from *in situ* sensors. We compare two approaches to this problem: (1) using forecasting models (Chapter 5) and (2) using feature vectors with extreme value theory (Chapter 4). In the forecasting models, observations are identified as outliers when they fall outside the bounds of an established prediction interval. Two strategies are considered for this comparison study: anomaly detection (AD) and anomaly detection and mitigation (ADAM) for the detection process. With ADAM, the detected outliers are replaced with the forecast prior to the next prediction, whereas AD simply uses the previous measurements without altering detected outliers. The feature-based framework first identifies the data features that differentiate outlying instances from typical behaviours. Then, statistical transformations are applied to make the outlying instances stand out in transformed data space. Unsupervised outlier scoring techniques are then applied to the transformed data space. An approach based on extreme value theory is used to calculate a threshold for each potential outlier. The proposed frameworks are evaluated using two datasets obtained from *in situ* sensors in rivers flowing into the Great Barrier Reef lagoon.

The feature-based approach (in Chapter 4) successfully identified outliers involving abrupt changes in turbidity, conductivity and river level, including sudden spikes, sudden isolated drops and level shifts, while maintaining very low false detection rates. It also has many special features: (1) It takes the correlation structure between the water-quality variables into account when detecting technical outliers, given some were apparent only in the high-dimensional space but not when each variable was considered independently. (2) It can be applied to both univariate and multivariate problems as the anomalous threshold is based either on the k nearest neighbour distances or the densities of data points. (3) Since this is an unsupervised algorithm, it can be easily extended to other water-quality variables, other sites and also to other outlier detection tasks in other application domains. The only requirement is to select suitable transformation methods according to the data features that differentiate the outlying instances from the typical behaviours of a given system. The transformations used in this study were mainly chosen as appropriate to the data collected from Sandy Creek and Pioneer River. Domain-specific knowledge plays a vital role when selecting suitable transformations. (4) Since the outlier threshold is derived from the spacing theorem from classical extreme value theory it has a valid probabilistic interpretation. 5) The use of derivative transformations allows it to deal with irregular(unevenly spaced) time series. (6) It can easily be extended to streaming data by incorporating a sliding window technique and then treating each window as a batch data set.

There is an important difference between the methods discussed in Chapter 4 and 5. Chapter 5 emphasizes the importance of different anomaly types and end-user needs and provides the starting point for constructing a framework for automated anomaly detection in high frequency water-quality data from *in situ* sensors. It emphasizes the use of forecasting models for detecting technical outliers in water quality data and considers the outlier detection problem in a univariate setting. It also briefly introduces unsupervised feature based methods for detecting technical-outliers in univariate data. Chapter 4 differs substantially from Chapter 5 as (1) the unsupervised feature based procedure for detecting technical-outliers in high frequency water-quality data measured by *in situ* sensors is its sole focus (2) the unsupervised feature based procedure is fully elaborated in both details and

depth and (3) the experimental results are enhanced through emphasis on the multivariate capabilities of the unsupervised feature based procedure. Since univariate and multivariate approaches have different focuses against different challenges direct comparison between the results of the two chapters were not made.

Furthermore, Chapter 4 focuses only on technical outliers involving abrupt changes in value, including sudden spikes, sudden isolated drops and level shifts (high priority outliers as described in Chapter 5) rather than the broader suite considered by Chapter 5. Owing to this difference between the focuses of the two chapters, different number of observations are considered as outliers in the two chapters. For example the 49 outliers in Pioneer River data in Chapter 4 only comprise abrupt changes including 3 large sudden spikes(type A), 5 sudden shifts (type D), 34 impossible values (type F) and 7 small sudden spikes (type J)). A detailed description for the different types of technical outliers are given in Table 2 in Chapter 5.

6.2 Future Work

Since this is a thesis by publication, each article should be self-contained and therefore has been published with all the relevant possible further research directions discussed in detail. To avoid repetition, this section summarises only the key future research priorities, which are deemed underrepresented in the current literature.

While the HDoutliers algorithm is powerful, several classes of counterexamples were identified where the structural properties of the data did not enable the HDoutliers algorithm to detect certain types of outliers. However, I acknowledge that these counterexamples are not diverse and challenging enough to generalise the findings to conclude that stray is always the superior algorithm. Therefore, an important open research problem is to assess the effectiveness of these algorithms across the broadest possible problem space defined by different datasets with diverse properties (Kang, Hyndman, and Smith-Miles, 2017). It would also be interesting to explore how other classes of problems with various structural properties can influence the performance of the stray algorithm and where its weaknesses might lie. This type of instance space analysis (Smith-Miles et al., 2014) will enable further insights into improved algorithm design.

In the oddstream algorithm, the use of the feature-based representation of time series is recommended, owing to its many advantages over the instance-based representation of time series. In the present study, only 14 features were used to represent a given time series. Further exploration of feature extraction and automatic feature selection methods is required to create a richer feature space that is suitable for many applications in the streaming data context. The proposed algorithm uses the first two principal components to obtain a two-dimensional feature space, and then defines an anomalous threshold on the resulting two-dimensional feature space. It is expected that in further studies, other dimension reduction techniques will be used, such as multidimensional scaling and random projection, to investigate the effects of such techniques on the performance of the proposed framework. Further, the density estimation in the proposed algorithm was performed using a bivariate kernel density estimation method. Since the density values in the tail are used to build the model of the typical behaviour, additional experiments need to be conducted on density estimation methods, to improve the tail estimation. On this topic, the log-spline bivariate density estimation method and the local likelihood density estimation method will be considered, with the aim of achieving a better tail estimation, and thereby improving the performance of the proposed algorithm. In the proposed algorithm, estimation of extreme value distributions of multivariate, multimodal mixture models requires sampling of extrema. Alternative methods as proposed in Hugueny, Clifton, and Tarassenko (2011) and Hugueny (2013) may guide further improvements in computational efficiency. They provide a light-weight formulation that they claim to be significantly faster than maximum-likelihood methods, which require large amounts of sampling. These alternative methods also provide solutions for multimodal multivariate models. Further, the current algorithm is developed under the assumption that the measurements produced by sensors are one-dimensional. Rapid advances in hardware technology have made it possible for many sensors to capture multiple measurements simultaneously, leading ultimately to a collection of multidimensional multivariate streaming time series data. Therefore, an important open research problem is to extend the oddstream algorithm to handle multidimensional, multivariate streaming data. Extending the oddstream algorithm to the detection of anomalies in this data context may allow us to perform anomaly detection in an even wider range of application domains.

Spatiotemporal anomaly detection for water-quality data also lags behind that for other *in situ* sensor data types (e.g., air quality or meteorology; Wu, Liu, and Chawla, 2008) because river data pose new challenges, such as the complex relationships between neighbouring sensors due to branching networks and flow directionality, tendency for biofouling and the highly dynamic nature of river water even under typical conditions (Kang et al., 2009). These challenges make traditional anomaly detection approaches inadequate for spatiotemporal water-quality data and require new methods. The oddwater algorithm is expected to expand into space and time so that it can deal with the spatiotemporal correlation structure along branching river networks. This will in turn provide a fundamental step-change in scientific understanding of the spatiotemporal dynamics of water quality in rivers and their networks and the potential downstream effects of pollutant loads.

6.3 Research Reproducibility

Research reproducibility is an important topic in modern science because it provides a general schema and an infrastructure to regenerate quantitative scientific results using the original datasets and methods (Stodden, Leisch, and Peng, 2014). Therefore, to facilitate reproducibility and reuse of the results presented in this thesis, I undertook several actions under the three key areas: software, data and papers (Stodden, Leisch, and Peng, 2014).

6.3.1 Software

This thesis introduces three R packages for anomaly detection.

The first R package is an accompaniment to the algorithm proposed in Chapter 2 and includes useful functions for detecting anomalies in high-dimensional data. Version 0.1.0 of the package was used for the results presented in Chapter 2 and is available from GitHub at `https://github.com/pridiltal/stray`.

The second R package, `oddstream`, is an accompaniment to the algorithm proposed in Chapter 3 and includes useful functions for detecting anomalous series within a large collection of streaming time series data. Version 0.5.0 of the package was used for the

results presented in Chapter 3 and is available from GitHub at https://github.com/pridiltal/oddstream.

The third package is an accompaniment to the algorithm proposed in Chapters 4 and 5 and includes useful functions for detecting technical anomalies in water-quality data from *in situ* sensors. Version 0.6.0 of the package was used for the results presented in Chapters 4 and 5 and is available from GitHub at https://github.com/pridiltal/oddwater.

6.3.2 Data

All the datasets on which the results are computed in each article are available via the corresponding R package. A Shiny web application available through the `oddwater` R package provides greater visual insight into the water-quality data from *in situ* sensors and was heavily used during the labelling process to pinpoint observations.

6.3.3 Papers

The three main articles in Chapters 2, 3 and 4 describe the corresponding algorithms in detail and compare their implementations using various datasets. The source files, including datasets and R code to reproduce all figures, tables and analysis of each article can be found in the following public GitHub repositories.

Chapter 2: 'Anomaly Detection for High Dimensional Data' at https://github.com/pridiltal/stray_manuscript.

Chapter 3: 'Anomaly Detection in Streaming Non-stationary Temporal Data' at https://github.com/pridiltal/oddstream_manuscript.

Chapters 4: 'A Feature-Based Procedure for Detecting Technical Outliers in Water-Quality Data from *in situ* Sensors' at https://github.com/pridiltal/oddwater_manuscript.

These articles were written entirely using `Rmarkdown` (Allaire et al., 2019), and compiled into a thesis using the `bookdown` R package (Xie, 2019), with the Monash PhD thesis rmarkdown template available at https://github.com/robjhyndman/MonashThesis. The source files of this thesis are available at https://github.com/pridiltal/PhD_Thesis_2019.

Bibliography

Abuzaid, A, A Hussin, and I Mohamed (2013). Detection of outliers in simple circular regression models using the mean circular error statistic. *Journal of Statistical Computation and Simulation* **83**(2), 269–277.

Aggarwal, CC (2017). *Outlier analysis*. Second edition. Cham, Switzerland : Springer.

Akoglu, L, H Tong, and D Koutra (2015). Graph based anomaly detection and description: a survey. *Data mining and knowledge discovery* **29**(3), 626–688.

Allaire, J, Y Xie, J McPherson, J Luraschi, K Ushey, A Atkins, H Wickham, J Cheng, W Chang, and R Iannone (2019). *rmarkdown: Dynamic Documents for R*. R package version 1.13. https://rmarkdown.rstudio.com.

Ben-Gal, I (2005). "Outlier detection". In: *Data Mining and Knowledge Discovery Handbook*. Springer, pp.131–146.

Burridge, P and AMR Taylor (2006). Additive outlier detection via extreme-value theory. *Journal of Time Series Analysis* **27**(5), 685–701.

Chandola, V, A Banerjee, and V Kumar (2009). Anomaly detection: A survey. *ACM Computing Surveys (CSUR)* **41**(3), 15.

Clifton, DA (2009). "Novelty detection with extreme value theory in jet engine vibration data". PhD thesis. University of Oxford.

Clifton, DA, S Hugueny, and L Tarassenko (2011). Novelty detection with multivariate extreme value statistics. *Journal of Signal Processing Systems* **65**(3), 371–389.

Coles, S (2001). *An Introduction to Statistical Modeling of Extreme Values*. Lecture Notes in Control and Information Sciences. Springer.

Embrechts, P, C Klüppelberg, and T Mikosch (2013). *Modelling Extremal Events: for Insurance and Finance*. Stochastic Modelling and Applied Probability. Springer Berlin Heidelberg.

Fanaee-T, H and J Gama (2016). Tensor-based anomaly detection: An interdisciplinary survey. *Knowledge-Based Systems* **98**, 130–147.

Faria, ER, IJ Gonçalves, AC de Carvalho, and J Gama (2016). Novelty detection in data streams. *Artificial Intelligence Review* **45**(2), 235–269.

Fisher, RA and LHC Tippett (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. 02. Cambridge Univ Press, pp.180–190.

Fulcher, BD and NS Jones (2014). Highly comparative feature-based time-series classification. *IEEE Transactions on Knowledge and Data Engineering* **26**(12), 3026–3037.

Fulcher, BD, MA Little, and NS Jones (2013). Highly comparative time-series analysis: the empirical structure of time series and their methods. *Journal of the Royal Society Interface* **10**(83), 20130048.

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

Galambos, J, J Lechner, and E Simiu (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.

Gama, J (2010). *Knowledge Discovery from Data Streams*. Chapman and Hall/CRC.

Grubbs, FE (1969). Procedures for detecting outlying observations in samples. *Technometrics* **11**(1), 1–21.

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

Habeeb, RAA, F Nasaruddin, A Gani, IAT Hashem, E Ahmed, and M Imran (2019). Real-time big data processing for anomaly detection: A Survey. *International Journal of Information Management* **45**, 289–307.

Hofmann, H, H Wickham, and K Kafadar (2017). Value plots: Boxplots for large data. *Journal of Computational and Graphical Statistics* **26**(3), 469–477.

Horsburgh, JS, SL Reeder, AS Jones, and J Meline (2015). Open source software for visualization and quality control of continuous hydrologic and water quality sensor data. *Environmental Modelling & Software* **70**, 32–44.

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

Hugueny, S, DA Clifton, and L Tarassenko (2011). Probabilistic Patient Monitoring with Multivariate, Multimodal Extreme Value Theory. In: *Biomedical Engineering Systems and Technologies*. Ed. by A Fred, J Filipe, and H Gamboa. Berlin, Heidelberg: Springer Berlin Heidelberg, pp.199–211.

Hyndman, RJ (1996). Computing and graphing highest density regions. *The American Statistician* **50**(2), 120–126.

Hyndman, RJ, E Wang, and N Laptev (2015). Large-scale unusual time series detection. In: *2015 IEEE international conference on data mining workshop (ICDMW)*. IEEE, pp.1616–1619.

Kang, JM, S Shekhar, M Henjum, PJ Novak, and WA Arnold (2009). Discovering teleconnected flow anomalies: A relationship analysis of dynamic neighborhoods (RAD) approach. In: *International Symposium on Spatial and Temporal Databases*. Springer, pp.44–61.

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

Kumar, D, JC Bezdek, S Rajasegarar, M Palaniswami, C Leckie, J Chan, and J Gubbi (2016). Adaptive cluster tendency visualization and anomaly detection for streaming data. *ACM Transactions on Knowledge Discovery from Data (TKDD)* **11**(2), 24.

Kwon, D, H Kim, J Kim, SC Suh, I Kim, and KJ Kim (2017). A survey of deep learning-based network anomaly detection. *Cluster Computing*, 1–13.

Lavin, A and S Ahmad (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.

Mitchell, R and IR Chen (2014). A survey of intrusion detection techniques for cyber-physical systems. *ACM Computing Surveys (CSUR)* **46**(4), 55.

Pimentel, MA, DA Clifton, L Clifton, and L Tarassenko (2014). A review of novelty detection. *Signal Processing* **99**, 215–249.

Pinto, C and P Garvey (2016). *Advanced Risk Analysis in Engineering Enterprise Systems*. Statistics: A Series of Textbooks and Monographs. CRC Press.

Ranshous, S, S Shen, D Koutra, S Harenberg, C Faloutsos, and NF Samatova (2015). Anomaly detection in dynamic networks: a survey. *Wiley Interdisciplinary Reviews: Computational Statistics* **7**(3), 223–247.

Schwarz, KT (2008). "Wind dispersion of carbon dioxide leaking from underground sequestration, and outlier detection in eddy covariance data using extreme value theory". PhD thesis.

Shahid, N, IH Naqvi, and SB Qaisar (2015). Characteristics and classification of outlier detection techniques for wireless sensor networks in harsh environments: a survey. *Artificial Intelligence Review* **43**(2), 193–228.

Singh, K and S Upadhyaya (2012). Outlier detection: applications and techniques. *International Journal of Computer Science Issues (IJCSI)* **9**(1), 307.

Smith-Miles, K, D Baatar, B Wreford, and R Lewis (2014). Towards objective measures of algorithm performance across instance space. *Computers & Operations Research* **45**, 12–24.

Stodden, V, F Leisch, and RD Peng (2014). *Implementing Reproducible Research*. CRC Press.

Wang, X, K Smith, and RJ Hyndman (2006). Characteristic-based clustering for time series data. *Data Mining and Knowledge Discovery* **13**(3), 335–364.

Weissman, I (1978). Estimation of parameters and large quantiles based on the k largest observations. *Journal of the American Statistical Association* **73**(364), 812–815.

Wilkinson, L (2018). Visualizing big data outliers through distributed aggregation. *IEEE Transactions on Visualization and Computer Graphics* **24**(1), 256–266.

Wu, E, W Liu, and S Chawla (2008). Spatio-temporal outlier detection in precipitation data. In: *International Workshop on Knowledge Discovery from Sensor Data*. Springer, pp.115–133.

Xie, Y (2019). *bookdown: Authoring Books and Technical Documents with R Markdown*. R package version 0.11. https://github.com/rstudio/bookdown.