Data Mining and Knowledge Discovery (2021) 35:1882–1905
https://doi.org/10.1007/s10618-021-00771-7

AURORA: A Unified fRamework fOR Anomaly detection on multivariate time series

Lin Zhang[1] · Wenyu Zhang[2] · Maxwell J. McNeil[1] · Nachuan Chengwang[1] · David S. Matteson[2] · Petko Bogdanov[1]

Received: 8 November 2019 / Accepted: 26 May 2021 / Published online: 23 June 2021

Abstract

The ability to accurately and consistently discover anomalies in time series is important in many applications. Fields such as finance (fraud detection), information security (intrusion detection), healthcare, and others all benefit from anomaly detection. Intuitively, anomalies in time series are time points or sequences of time points that deviate from normal behavior characterized by periodic oscillations and long-term trends. For example, the typical activity on e-commerce websites exhibits weekly periodicity and grows steadily before holidays. Similarly, domestic usage of electricity exhibits daily and weekly oscillations combined with long-term season-dependent trends. *How can we accurately detect anomalies in such domains while simultaneously learning a model for normal behavior?* We propose a robust offline unsupervised framework for anomaly detection in seasonal multivariate time series, called AURORA. A key innovation in our framework is a general background behavior model that unifies periodicity and long-term trends. To this end, we leverage a Ramanujan periodic dictionary and a

Responsible editor: Ira Assent, Carlotta Domeniconi, Aristides Gionis, Eyke Hüllermeier.

✉ Lin Zhang
lzhang22@albany.edu

Wenyu Zhang
wz258@cornell.edu

Maxwell J. McNeil
mmcneil2@albany.edu

Nachuan Chengwang
nchengwang@albany.edu

David S. Matteson
matteson@cornell.edu

Petko Bogdanov
pbogdanov@albany.edu

1 Department of Computer Science, University at Albany—SUNY, Albany, NY, USA

2 Cornell University, Statistics and Data Science, Ithaca, NY, USA

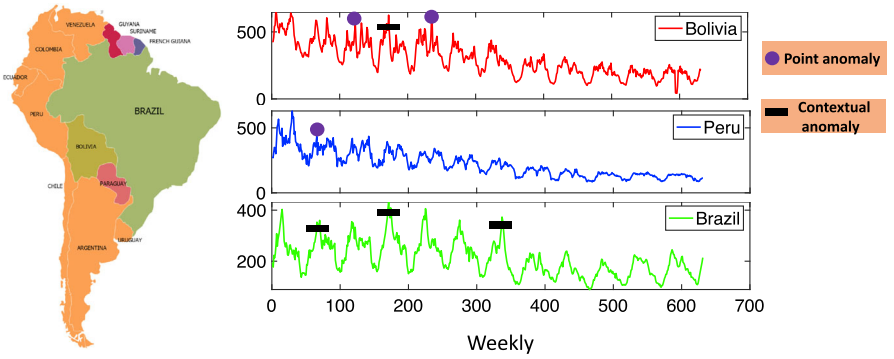
spline-based dictionary to capture both seasonal and trend patterns. We conduct experiments on both synthetic and real-world datasets and demonstrate the effectiveness of our method. AURORA has significant advantages over existing models for anomaly detection, including high accuracy (AUC of up to 0.98), interpretability of recovered normal behavior (100% accuracy in period detection), and the ability to detect both point and contextual anomalies. In addition, AURORA is orders of magnitude faster than baselines.

Keywords Offline anomaly detection · Multivariate time series · Periodic dictionary · Spline dictionary · Alternating optimization

1 Introduction

Multivariate time series data arise in many domains including the web (Xu et al. 2018b), sensor networks (Zhang et al. 2019a), database systems (Van Aken et al. 2017), finance (Zhu and Shasha 2002) and others. Time series from the above domains often exhibit both seasonal behavior and long-term trends (Saad et al. 1998; Luo et al. 2005). For example, city traffic levels (Jindal et al. 2013) and domestic energy consumption (Hong et al. 2016) both have an inherent period related to the regularity of human activity (daily/weekly oscillations) and long term trends. Consider, for example, Google flu searches (Google 2014) over 13 years at one week granularity visualized in Fig. 1. Search rates exhibit annual seasonality, while in the long term neighboring countries share a decreasing trend. In this type of time series, a common problem across domains is the detection of anomalous time points in each of the co-evolving time series. *How to effectively detect anomalous time points in seasonal time series with long-term trends without prior knowledge and supervision?*

Accurate anomaly detection enables a host of applications including health monitoring and risk prevention (Vallis et al. 2014), financial fraud loss protection (Goldstein



(a) Map **(b)** Time series

Fig. 1 An example of multivariate time series with seasonality, trends, and different types of anomalies: **a** The map shows three neighbouring countries in south America: Peru, Brazil and Bolivia; **b** the graph shows weekly time series of Google flu searches for these three countries spanning from 2002 to 2015. Point and segment annotations are predicted anomalies

2014) and data cleaning (Zhang et al. 2017). Anomalies are defined as "points" that deviate significantly from the normal state and are differentiated into two types: point and contextual anomalies. Point anomalies, denoted as circles in Fig. 1, consist of a single outlying observation which stands out. Contextual anomalies (segments in Fig. 1) consist of sequences of outliers, which can be mistaken for normal behavior due to their persisting nature, and hence require long-term contextual information to be detected. The detection of anomalies in multivariate time series elucidates the behavior of a system holistically—both normal and abnormal. For instance, it can help us make sense of patterns and anomalies in the co-evolving national time series of cases of COVID-19.

Anomaly detection is often formulated as a classification problem (Liu et al. 2015; Laptev et al. 2015), and as such, it requires supervised model learning. However, the labels for anomalies are rare and typically expensive to obtain, leading to a surge of interest in unsupervised approaches, including statistical methods (Manevitz and Yousef 2001; Hautamaki et al. 2004; Breunig et al. 2000) and deep-learning (An and Cho 2015; Zhang et al. 2019a). While some existing methods do not rely on explicit anomaly annotations, they have three fundamental limitations: (1) they require normal (non-anomalous) data, (2) they are sensitive to small local variations, and (3) they do not offer model interpretability.

In this work, we propose an unsupervised offline (or batch) method, called AURORA, to detect anomalies in multivariate time series with an explicit estimation of the normal behavior as a mixture of seasonality and trends. Our formulation of normal behavior is flexible and allows capturing diverse temporal patterns. Namely, we formulate seasonality and trend fitting as an optimal reconstruction problem employing the Ramanujan periodic dictionary and a spline dictionary, respectively. This new framework ensures the accurate discovery of interpretable normal behavior while highlighting anomalies.

Our contributions in this paper are as follows:

Novel formulation We introduce the problem of anomaly detection in seasonal multivariate time series with long-term non-linear trends.

Interpretability Our framework AURORA automatically detects anomalies and simultaneously obtains an interpretable model of the seasonal and trend components in the observed time series.

Applicability AURORA ensures strong quantitative improvements on synthetic and real-world datasets in anomaly detection (up to 30% improvement on AUC), and superior scalability (14 seconds in data with half a million time points and 200 univariate series) when compared to state of the art baselines.

2 Related work

2.1 Anomaly detection

Existing methods can be broadly categorized into supervised and unsupervised. Supervised anomaly detection methods such as one-class SVM (Manevitz and Yousef 2001) and Isolation Forest (Liu et al. 2008) employ labeled anomaly data and pose the

problem as binary classification (Aminikhanghahi and Cook 2017). Since labels for anomalies in time series are expensive to obtain and largely unavailable (Zhang et al. 2019a), we focus on the unsupervised case. In unsupervised settings, distance-based methods, such as kNN (Hautamaki et al. 2004), are commonly used to quantify the difference between normal and anomalous samples. Subspace learning methods have also been proposed where the goal is to identify a subspace in which the difference between normal and anomalous samples are more pronounced (Keller et al. 2012). Both distance-based and subspace-based methods are designed for anomaly detection in datasets of independent samples and do not consider the temporal structure of time series. Thus, they are not well-suited to localize anomalies in time series.

To account for the temporal structure, some methods explicitly model temporal patterns while others utilize comparisons across time. In the former category normal behavior with multiple periods is rarely considered and many time series models are restricted to the univariate case. The Autoregressive Moving Average framework (Chen and Liu 1993) accounts for temporal correlations and can be extended to include seasonality, but is sensitive to noise and is limited to single-period time series. TwitterR (Hochenbaum et al. 2017) applies the Extreme Studentized Deviate outlier test after decomposing the univariate time series into its median, seasonal and residual component. This method assumes that periodicity is known and only allows a single period. Other methods map the time series in some feature space (Chan and Mahoney 2005), a process which requires domain knowledge to construct informative features. Methods relying on comparison across time declare a time window as anomalous if its distance to past windows is too large (Wei et al. 2005). However, it is challenging to select an optimal window size for the analysis. Such methods also requires some supervision in that it relies on a sanitized (anomaly-free) period of observations. Matrix profile (Yeh et al. 2016) is another popular distance-based method. It profiles the distance of all fixed-length subsequences to their top-1 closest subsequence. Since Matrix Profile operates at the subsequence level, and thus can declare only windows of pre-selected size as anomalous, it is not a good fit for our task of pinpointing individual anomaly time points or contextual anomalies of variable length (De Paepe et al. 2019).

Given the wide variety of structures in multivariate time series, recently deep learning techniques have gained increasing attention in anomaly detection (Zhang et al. 2019a; Hundman et al. 2018; An and Cho 2015; Xu et al. 2018a; Zhou and Paffenroth 2017). In Zhang et al. (2019a), the authors developed a multi-scale model that captures the temporal patterns. Authors of Hundman et al. (2018) compare LSTM model predictions to actual observations and use thresholding to detect anomalies. Several *variational autoencoder (VAE)* methods have also been proposed for this task (An and Cho 2015; Xu et al. 2018a; Zhou and Paffenroth 2017). The recent meta-analysis by Zhang et al. (2019a) demonstrates that such deep learning methods do not exploit the temporal structure, suffer from sensitivity to noise, and do not offer interpretation. In addition, deep learning anomaly detectors also require normal (anomaly-free) data to train the underlying "normal" model. Therefore, these methods are not applicable to fully unsupervised scenarios where normal data is not available. As we demonstrate experimentally, our method consistently outperforms representative methods from this group, even if they are given access to normal data in both synthetic and real-world applications.

2.2 Change point detection in time series

Change points can be viewed as a specific type of anomaly where the change is long-lasting in nature. Statistical change point detection methods generally assume independent and identically distributed data within a temporal segment to establish probabilistic models, and cannot be applied directly on data with periodicity and trends. They require multiple observations in each segment for accurate estimation (Zhang et al. 2017), and hence may be limited to detecting contextual anomalies. Some methods require knowledge of the underlying data distributions (Killick et al. 2012), while others constrain the target change point types, for instance focusing on level shifts (Bleakley and Vert 2011; Zhang et al. 2019b).

2.3 Period learning

Traditional period detection methods have employed Fourier transform (Li et al. 2010; Indyk et al. 2000) and work in the frequency domain to determine pronounced periods. The major drawback of such methods is the detection of a large number of spurious periods (Tenneti and Vaidyanathan 2015). Auto-correlation is another approach for period learning (Davies and Plumbley 2005) which employs similarity among time segments. Methods in this category rely on a predefined threshold for selecting dominant periods and often employ heuristic post-processing or integration with Fourier spectrograms (Vlachos et al. 2005). Recently, a dictionary-based period learning framework has been proposed by Tenneti and Vaidyanathan (2015), comprised of a family of periodic dictionaries and a unified convex model for period detection. Recent work has offered improvements by exploiting the group structure in the dictionary (Zhang et al. 2020; Zhang and Bogdanov 2020).

Period learning methods were also proposed for binary sequence data (Li et al. 2015; Yuan et al. 2017). These methods assume that the time series have only one period and rely on prior information about the series. For example, the model in Li et al. (2015) requires an appropriate segmentation threshold. More importantly, these methods are only applicable to binary sequences and are not suited to deal with general time series.

3 Preliminaries and notation

3.1 Notation

We denote by A_i, A^i, A_{ij} the i-th column, the i-th row, and the ij-th element of matrix A, respectively. Throughout the paper, $\|\cdot\|_F$, $\|\cdot\|_1$ and $\|\cdot\|_*$ denote the Frobenius, L1 and nuclear norms. The nuclear norm is defined as $\|A\|_* = \sum_i \sigma_i$, where σ_i is the i-th singular value of A. I denotes the identity matrix.

3.2 The Ramanujan periodic dictionary

The Ramanujan periodic dictionary was proposed by Tenneti and Vaidyanathan (2015) to learn underlying periods in univariate time series. For a given period g, the Ramanujan periodic dictionary is defined as a nested matrix:

$$\Phi_g = \left[P_{d_1}, P_{d_2}, \ldots P_{d_K}\right], \tag{1}$$

where $\{d_1, d_2, \ldots, d_K\}$ are the divisors of the period g sorted in an increasing order; $P_{d_i} \in \mathbb{R}^{g \times \phi(d_i)}$ is a periodic basis matrix for period d_i, where $\phi(d_i)$ denotes the Euler totient function evaluated at d_i and $d = \sum_i \phi(d_i)$. The basis matrix here is constructed based on the Ramanujan sum (Tenneti and Vaidyanathan 2015):

$$C_{d_i}(g) = \sum_{k=1, gcd(k,d_i)=1}^{d_i} e^{j2\pi kg/d_i}, \tag{2}$$

where $gcd(k, d_i)$ denotes the greatest common divisor of k and d_i. The Ramanujan periodic basis matrix is constructed as a circulant matrix as follows:

$$P_{d_i} = \begin{bmatrix} C_{d_i}(0) & C_{d_i}(g-1) & \ldots & C_{d_i}(1) \\ C_{d_i}(1) & C_{d_i}(0) & \ldots & C_{d_i}(2) \\ \ldots & \ldots & \ldots & \ldots \\ C_{d_i}(g-1) & C_{d_i}(g-2) & \ldots & C_{d_i}(0) \end{bmatrix}. \tag{3}$$

To this end, we can obtain the overall Ramanujan periodic dictionary R for maximum period g_{max} as $R = \left[\Phi_1, .., \Phi_{g_{max}}\right]$, where R is also called a nested periodic matrix (NPM). To analyze time series of length T, the columns in R are periodically extended to a length of T. One can reconstruct time series as a linear combination of a few columns from R, where the dominant periods of time series correspond to high-magnitude coefficients.

3.3 Spline dictionary

PB-spline regression (Goepp et al. 2018; Eilers and Marx 2010) is a flexible method to fit curves using B-splines of degree-d with a smoothness regularization. Quadratic or cubic B-splines are sufficient for most applications (Eilers and Marx 2010). A spline of degree d with k distinct interior knots $\{u_1, \ldots, u_k\}$ is a function constructed by connecting and summing polynomial segments of degree d. We construct the spline from B-splines basis functions $B_{i,d}(u)$ which can be defined recursively by the Cox-de-Boor formula: $B_{i,0} = 1$ if $u_i \leq u < u_{i+1}$, 0 otherwise.

$$B_{i,p} = \frac{u - u_i}{u_{i+p} - u_i} B_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} B_{i+1,p-1}(u)$$

Each $B_{i,d}(u)$ is non-zero on $[u_i, u_{i+d+1})$. The resulting spline is a linear combination of the basis functions. A sequence of equally-spaced knots is often specified, with a regularization term to penalize for overfitting and to encourage smoothness of the fitted curve.

4 Problem definition

In this paper we study the anomaly detection problem on multivariate time series. Suppose we are given a multivariate time series matrix $X \in \mathbb{R}^{t \times n}$ with t time points and n samples. We model this input as a mixture of three components:

$$X = X_T + X_S + O + \delta, \tag{4}$$

where X_S denotes the seasonal component, X_T denotes the trend component, O represents anomalies, and δ is random noise. In general, prior information about these three components is not available, therefore, we propose to minimize the following objective to model them:

$$\underset{X_S, X_T}{\operatorname{argmin}} \quad \|X - X_S - X_T\|_1 + R_1(X_S) + R_2(X_T), \tag{5}$$

where anomalies are computed as the residual: $O = X - X_S - X_T$. In the above objective, $R_1(X_S)$ and $R_2(X_T)$ are regularization terms for X_S and X_T, respectively. Note that minimizing the L1 norm reconstruction cost is robust to anomalies, therefore, the objective can capture the X_T and X_S components without being sensitive to distortion. The two learned components comprise the normal temporal behavior of the data.

4.1 Seasonality modeling $R_1(X_S)$

We impose structure on the periodic component by harnessing the Ramanujan periodic dictionary. Namely, we convert this problem as a sparse coding problem into follows,

$$\underset{U}{\operatorname{argmin}} \quad \lambda_1 \|U\|_1, \quad s.t. \quad X_S = GU \tag{6}$$

where $G = RH^{-1} \in \mathbb{R}$ and λ_1 is a balance parameter. Here, H is a diagonal matrix for penalizing large periods in R when sparse coding, where $H_{ii} = p^2$ and p is the period of the i-th column in the R. Finally, the coefficients of periods can be obtained through $\hat{U} = H^{-1}U$.

4.2 Trend modeling $R_2(X_T)$

Our second goal is to impose structure on the trend component X_T. Trends do not typically follow a parametric regular shape and users have no prior information about

it, therefore, we employ a spline-based approximation, which is an effective nonparametric smooth shape estimation solution based on polynomial functions. We introduce a spline dictionary, denoted as $A \in \mathbb{R}^{t \times m}$, where each column represents a spline basis function. We can use the degree-d B-spline basis defined over k internal knots, such that $m = k + d + 1$. We employ equally-spaced knots to construct the dictionary. To ensure separability between the periodic and trend components, we pre-process the spline basis S to be orthogonal to the periodic dictionary R. Let $\tilde{R}$ be an orthonormal version of R, which can be constructed by the Gram-Schmidt process. $\tilde{R}$ has the same dimensions as R since the latter forms a basis. We can then find the component of S perpendicular to the subspace spanned by the columns of $\tilde{R}$ by $A = S - \tilde{R}\tilde{R}'S$.

By treating A as the underlying factors, X_T can be linearly reconstructed in terms of these factors as follows:

$$\underset{W}{\text{argmin}} \quad \lambda_2 \|W\|_* + \lambda_3 \|DW\|_F^2 \,, s.t. \quad X_T = AW, \tag{7}$$

where W is the coefficient matrix and D is the matrix form of the difference operator; and λ_2 and λ_3 are balance parameters. Multivariate time series are often collected from the same system, thus they often share similar global trends. To this end, we impose a low-rank constraint on W through a nuclear norm penalty. We also incorporate a selection regularization on W as $\|DW\|_F^2$ to prevent overfitting and to encourage smoothness of the fitted curve. We define this difference penalty by introducing the cases of 1st $\sim$ 3rd order difference as follows:

$$\begin{cases} [DW]_{ij} = W_{ij} - W_{i-1,j} & \text{1th-order} \\ [DW]_{ij} = W_{ij} - 2W_{i-1,j} + W_{i-2,j} & \text{2th-order} \\ [DW]_{ij} = W_{ij} - 3W_{i-1,j} + 3W_{i-2,j} - W_{i-3,j} & \text{3th-order.} \end{cases} \tag{8}$$

We use 3^{rd} order as the default setting, except when noted otherwise. Note that the selection of the type of spline basis functions in dictionary A is flexible, but certain choices such as truncated polynomials are known to be prone to numerical instability.

4.3 Overall AURORA objective

By integrating all the above, we arrive at the objective function for anomaly detection on multivariate time series as follows:

$$\underset{U,W}{\text{argmin}} \quad \|X - GU - AW\|_1 + \lambda_1 \|U\|_1 + \lambda_2 \|W\|_* + \lambda_3 \|DW\|_F^2 \,. \tag{9}$$

Instead of modeling outliers explicitly, we detect anomalies from the residuals of $O = X - GU - AW$. Intuitively, anomalies diverge from normal components, i.e. seasonal and trend component, and thus, lead to large residuals. We produce a ranked list of possible anomaly locations corresponding to high magnitude of the residuals. Without any assumptions on anomaly lengths, this model allows us to detect any

deviation from the normal state, instead of being restricted to only detecting certain types of anomalies.

5 Optimization

Since the objective function in Eq. 9 is not jointly convex, we optimize the two variables alternatively using the Alternating Direction Method of Multiplier (ADMM) framework (Boyd et al. 2011). We first introduce auxiliary variables: $V = U$, $P = W$ and $Y = X - GU - AW$. Then, we rewrite Eq. 9 as follows:

$$\begin{aligned} &\underset{U,W,V,P,Y}{\operatorname{argmin}} \quad \|Y\|_1 + \lambda_1 \|V\|_1 + \lambda_2 \|P\|_* + \lambda_3 \|DW\|_F^2 \\ &s.t \quad V = U, P = W, Y = X - GU - AW. \end{aligned} \tag{10}$$

The corresponding Lagrangian function is:

$$\begin{aligned} &\mathcal{L}(U, W, V, P, Y, \Lambda_1, \Lambda_2, \Lambda_3) = \|Y\|_1 + \lambda_1 \|V\|_1 + \lambda_2 \|P\|_* + \lambda_3 \|DW\|_F^2 \\ &+ \langle \Gamma_1, V - U \rangle + \frac{\rho_1}{2} \|V - U\|_F^2 + \langle \Gamma_2, P - W \rangle + \frac{\rho_2}{2} \|P - W\|_F^2 \\ &+ \langle \Gamma_3, Y - (X - GU - AW) \rangle + \frac{\rho_3}{2} \|Y - (X - GU - AW)\|_F^2 \end{aligned} \tag{11}$$

where $\Gamma_1 \sim \Gamma_3$ are the Lagrangian multipliers and $\rho_1 \sim \rho_3$ are penalty parameters.

5.1 Update *Y*

The subproblem w.r.t. V is as follows:

$$\underset{Y}{\operatorname{argmin}} \quad \|Y\|_1 + \frac{\rho_3}{2} \left\| Y - (X - GU - AW) + \frac{\Gamma_3}{\rho_3} \right\|_F^2 \tag{12}$$

This problem can be solved based on the following Lemma from Lin et al. (2013).

Lemma 1 *For $\alpha > 0$, the following objective has a closed-form solution*

$$\underset{A}{\operatorname{argmin}} \quad \frac{1}{2} \|A - B\|_F^2 + \alpha \|A\|_1 \tag{13}$$

which is written as $A_{ij} = sign\left(B_{ij}\right) \times max\left(\left|B_{ij}\right| - \alpha, 0\right)$. Here, $sign(t)$ is the signum function defined as:

$$sign(t) = \begin{cases} 1 & if\, t > 0 \\ -1 & if\, t < 0 \\ 0 & if\, t = 0 \end{cases} \tag{14}$$

Based on this lemma, we obtain a closed-form solution for Y:

$$Y_{ij} = sign\left(E_{ij}\right) \times max\left(\left|E_{ij}\right| - \frac{1}{\rho_3}, 0\right) \tag{15}$$

where $E = (X - GU - AW) - \frac{\Gamma_3}{\rho_3}$.

5.2 Update *V*

The subproblem w.r.t. V is as follows:

$$\underset{V}{\text{argmin}} \quad \lambda_1 \|V\|_1 + \frac{\rho_1}{2} \left\| V - U + \frac{\Gamma_1}{\rho_1} \right\|_F^2 \tag{16}$$

We similarly obtain a closed-form solution for V employing Lemma 1:

$$V_{ij} = sign\left(H_{ij}\right) \times max\left(\left|H_{ij}\right| - \frac{\lambda_1}{\rho_1}, 0\right), \tag{17}$$

where $H = U - \frac{\Gamma_1}{\rho_1}$.

5.3 Update *P*

The subproblem w.r.t. P is as follows:

$$\underset{P}{\text{argmin}} \quad \lambda_2 \|P\|_* + \frac{\rho_2}{2} \left\| P - W + \frac{\Gamma_2}{\rho_2} \right\|_F^2 \tag{18}$$

According to the singular value thresholding (SVT) method (Cai et al. 2010), we can compute a closed-form solution for P as well. By setting $M = W - \frac{\Gamma_2}{\rho_2}$, we first take the singular value decomposition of M as $M = J\Sigma K^T$, where J, K and Σ denote left-singular vectors, right-singular vectors, and singular values, respectively. Then, we obtain the solution for P as $P = J\mathcal{S}(\Sigma)K^T$, where $\mathcal{S}(\Sigma) = diag\left[max\left(\sigma_i - \frac{\lambda_2}{\rho_2}, 0\right)\right]$ and σ_i is the ith singular value.

5.4 Update *U*

The subproblem w.r.t. U is:

$$\underset{U}{\text{argmin}} \quad \frac{\rho_1}{2} \left\| V - U + \frac{\Gamma_1}{\rho_1} \right\|_F^2 + \frac{\rho_3}{2} \left\| Y - (X - GU - AW) + \frac{\Gamma_3}{\rho_3} \right\|_F^2 \tag{19}$$

By taking the gradient w.r.t. U and equating it to zero, we obtain:

$$\rho_1 U - (\rho_1 V + \Gamma_1) + \rho_3 G^T \left(GU - X + AW + Y + \frac{\Gamma_3}{\rho_3} \right) = 0 \tag{20}$$

We get the closed-form solution of U as follows:

$$U = \left(\rho_1 I + \rho_3 G^T G \right)^{-1} \left[\rho_1 V + \Gamma_1 + \rho_3 G^T \left(X - AW - Y - \frac{\Gamma_3}{\rho_3} \right) \right] \tag{21}$$

5.5 Update *W*

The subproblem w.r.t. W is as follows:

$$\underset{W}{\text{argmin}} \quad \lambda_3 \|DW\|_F^2 + \frac{\rho_2}{2} \left\| P - W + \frac{\Gamma_2}{\rho_2} \right\|_F^2 + \frac{\rho_3}{2} \left\| Y - (X - GU - AW) + \frac{\Gamma_3}{\rho_3} \right\|_F^2 \tag{22}$$

By setting the above objective's derivative w.r.t. W to zero, we obtain:

$$\lambda_3 D^T DW + \rho_2 \left(W - P - \frac{\Gamma_2}{\rho_2} \right) + \rho_3 A^T \left(AW - GU + Y + \frac{\Gamma_3}{\rho_3} \right) = 0 \tag{23}$$

As a result, a closed-form solution for W is as follows:

$$W = \left(\lambda_3 D^T D + \rho_2 I + \rho_3 A^T A \right)^{-1} \left[\rho_2 P + \Gamma_2 + \rho_3 A^T \left(GU - Y - \frac{\Gamma_3}{\rho_3} \right) \right] \tag{24}$$

Updates for the Lagrangian multipliers Γ_1, Γ_2 and Γ_3: In the $i+1$ iteration, the Lagrangian multipliers can be updated as follows: $\Gamma_1^{i+1} = \Gamma_1^i + \rho_1 (V - U)$, $\Gamma_2^{i+1} = \Gamma_2^i + \rho_2 (P - W)$, and $\Gamma_3^{i+1} = \Gamma_3^i + \rho_3 [Y - (X - GU - AW)]$.

5.6 Overall algorithm and complexity analysis

We summarize AURORAin Algorithm 1. We repeatedly perform the updates for U and W from Steps 3 to Step 12 until convergence. The most substantial running time cost is due to Steps 5, 7 and 8, while the remaining steps are either of linear complexity or near-linear complexity, e.g. involvivng sparse matrix multiplication. For Step 5, the SVD operation has a complexity of $O\left(min\left(tn^2, t^2n\right)\right)$. Here, t is often much greater than n, therefore, the complexity of svd is $O\left(tn^2\right)$. Both step 7 and 8 involve an inversion of a quadratic matrix, which incurs a cost of $O(q^3)$ and $O(m^3)$ in the worst case, respectively. Because of the sparsity in D and I, the complexity of the two can be significantly reduced to $O(qS_{nnz})$ and $O(mS_{nnz})$ based on the analysis in Zhang and Bogdanov (2019). In the above, S_{nnz} is the number of non-zero elements of

Algorithm 1 AURORA

Require: A multivariate time series matrix X, and parameters ($\lambda_1 \sim \lambda_3$).
1: Initialize: $\Gamma_1 = 0, \Gamma_2 = 0, \Gamma_3 = 0$; $\rho_1 = 1, \rho_3 = 1, \rho_3 = 1$.
2: **while** W and U have not converged **do**
3: $Y_{ij} = sign\left(E_{ij}\right) \times max\left(\left|E_{ij}\right| - \frac{1}{\rho_2}, 0\right)$
4: $V_{ij} = sign\left(H_{ij}\right) \times max\left(\left|H_{ij}\right| - \frac{\lambda_1}{\rho_1}, 0\right)$
5: $[J, \Sigma, K] = svd(W - \frac{\Gamma_2}{\rho_2})$
6: $P = J\mathcal{S}\left(\Sigma\right) K^T$
7: $U = \left(\rho_1 I + \rho_3 G^T G\right)^{-1}\left[\rho_1 V + \Gamma_1^i + \rho_3 G^T\left(X - AW - Y - \frac{\Gamma_3^i}{\rho_3}\right)\right]$
8: $W = \left(\lambda_3 D^T D + \rho_2 I + \rho_3 A^T A\right)^{-1}\left[\rho_2 P + \Gamma_2^i + \rho_3 A^T\left(GU - Y - \frac{\Gamma_3^i}{\rho_3}\right)\right]$
9: $\Gamma_1^{i+1} = \Gamma_1^i + \rho_1\left(V - U\right)$
10: $\Gamma_2^{i+1} = \Gamma_2^i + \rho_2\left(P - W\right)$
11: $\Gamma_3^{i+1} = \Gamma_3^i + \rho_3\left[Y - (X - GU - AW)\right]$
12: $i = i + 1$
13: **end while**
14: $O = X - GU - AW$
15: **return** $\{O, W, U\}$

$\lambda_3 D^T D + \rho_2 I$ and I_{nnz} is the number of non-zero elements of I. Therefore, the overall asymptotic complexity for each iteration is $O\left(tn^2 + qI_{nnz} + mS_{nnz}\right)$. In practice, AURORA often only needs a few iterations to converge. An implementation of our method is available for download at http://www.cs.albany.edu/~petko/lab/code.html.

6 Experimental evaluation

Our goal is to evaluate the performance of our method on data with periodic and trend components with both single point and contextual anomalies. Thus, our selection of datasets, baselines and evaluation metrics are geared towards this setting.

6.1 Datasets

We conduct evaluation experiments on both synthetic and real-world datasets.

6.1.1 Synthetic data

We generate 20-dimensional time series of length 5000 and refer to each univariate time series as a sample. The time series are comprised of four additive components: (1) *periodic*, (2) *trend*, (3) *anomalies* and (4) *noise* components. We generate the *periodic component* by following the methodology in Tenneti and Vaidyanathan (2015), namely, we generate a uniformly random sequence of length equal to a pre-specified underlying period and repeat it to obtain a series of the desired length. We select two periods for each time series randomly from {3, 5, 7, 11, 13}. We use 4-th degree polynomial functions to generate the *trends* in the time series with two sets of coeffi-

cients: $\{[1, 1, 0, -0.1], [1, 0.1, 0.1, 0.1]\}$. Individual samples are assigned one of the two trend polynomials randomly.

We select random individual time points as positions for *point anomalies* and add random values to the normal behavior of varying magnitude as outlined in individual experiments. To inject *contextual anomalies*, we select a position uniformly at random in the time series and add contiguous point anomalies (as described above), of length chosen uniformly from $\{3, 4, 5, 6\}$. All contextual anomalies are independent in each univariate (sample) time series. We also add *Gaussian noise* to all time series and control the signal-to-noise ratio (SNR) in the experiments.

6.1.2 Real-world data

We also experiment with time series from 4 real-world datasets, including *Power plant* (Rosca et al. 2015) and *Google flu* (Google 2014), Yahoo (Laptev and Amizadeh 2015) and NAB (Lavin and Ahmad 2015). We inject point anomalies in the *Power plant* (Rosca et al. 2015) and *Google flu* (Google 2014) following the same protocol as in our synthetic datasets. Note that this is a common evaluation strategy when anomaly labels are not available (Rosca et al. 2015; Emmott et al. 2015). The rest of the datasets have anomaly labels.

• *Power plant* (Rosca et al. 2015): This dataset is from the 2015 PHM Society Data Challenge. There are a total of 24 sensors with a sampling rate of 15 minutes. We experiment with the first segment of 10,000 time steps. We randomly inject 6 point anomalies for each sensor (144 in total) by following our synthetic anomaly injection methodology. Anomaly positions are independent in each sample.

• *Google flu* (Google 2014): The Google flu dataset consists of weekly estimates for influenza rates based on web searches in 29 countries from 2002 to 2015 (659 time points). We inject 6 point anomalies in the time series of each country.

• *Yahoo* (Laptev and Amizadeh 2015): The Yahoo A1 benchmark has 67 time series with labeled anomalies. This is a collection of real traffic metrics from Yahoo! services reported hourly. The lengths of individual time series varies between 741 and 1461. Note that existing literature argues that ground truth in some time series are not reliable (De Paepe et al. 2020). We have annotated such time series in the experiments, but report results on all of them for completeness.

• *NAB* (Lavin and Ahmad 2015): The NAB dataset includes time series data from multiple domains with manually labelled anomalies. We report results on the 10 Twitter traffic time series from the benchmark as they fit our assumptions of periodic behavior.

6.2 Experimental setup

6.2.1 Baselines

• *Anomaly detection* We compare AURORA on anomaly detection with two baselines: TwitterR (Hochenbaum et al. 2017) and Donut (Xu et al. 2018b). These state-of-the-art anomaly detectors are both flexible in detecting time series patterns of varying length. While Matrix profile (Yeh et al. 2016) is another potential baseline, it operates at a

fixed-length subsequence level and is not applicable to detecting anomalies of variable length. Hence, we do not employ it for comparison.
Donut (Xu et al. 2018b) is based on the Variational Autoencoder (VAE) framework. It detects anomalies by scoring dependencies in a time window of a fixed length based on a pre-trained model for anomalies. This method accounts for periodicity and trends in the time series and is, thus, an especially good fit for our setting. The window size is the main parameter in the method. We report the best result based on a grid search on window sizes varying from 10 to 100 with a step of 10. In all experiments a window size with 10 resulted in the best performance.
TwitterR (Hochenbaum et al. 2017) employs the Seasonal Extreme Studentized Deviate (S-ESD), a popular and robust anomaly detection method for univariate seasonal data. Raw data is decomposed into a median component, seasonal component and residuals. The Extreme Studentized Deviate (ESD) test is then applied to the residuals to produce a list of anomaly time points ordered by their probability of being an outlier. Different from our method which learns the underlying periods from data, this algorithm requires a single dominant period as an input. We set this parameter to the minimum true period present in our synthetic time series which puts the method in an advantageous position. We are, however, interested in characterising its quality for anomaly detection in the best possible settings, and thus, control for other factors of inaccuracy. For experiments on real-world data, the true periods are unknown, so we set TwitterR's period parameter according to the highest-magnitude DFT frequency in each time series.
• *Period detection* AURORA learns the underlying periods from data, and hence, we also evaluate its ability to detect GT periods and compare to three baselines:
NPM (Tenneti and Vaidyanathan 2015) is the state-of-the-art period learning method based on periodic dictionaries. It encodes time series employing the Ramanujan periodic dictionary and predicted periods are recovered based on the reconstruction coefficients. Since NPM operates on univariate time series, we apply it on each univariate time series and compile top ranked periods as final predictions.
FFT (Priestley 2004) is a classical period learning method that transforms time series into their frequency domain. Predicted periods have high-magnitude coefficients.
AUTO (Li et al. 2015) combines auto-correlation and Fourier transform. It first calculates the auto-correlation of the input data. Next, it employs Fourier transform on the results from the first step to derive periods of highest magnitude.

6.2.2 Performance metrics

We employ area under the ROC curve (AUC) to quantify the performance in anomaly detection. A true positive (TP) in the case of point anomalies is the correct identification of a time point annotated as a ground truth anomaly. We treat contextual (interval) anomalies as a collection of point anomalies (e.g., a contextual anomaly of length l is treated as l positive instances of anomalies), and evaluate AUC for this case in the same manner. Note that this measure is naturally biased to longer contextual anomalies which correspond to positive examples proportional to their length. Since, in some experiments we consider both point and contextual anomalies in the same time series, we focus on this simple measure that can capture both types. It is also

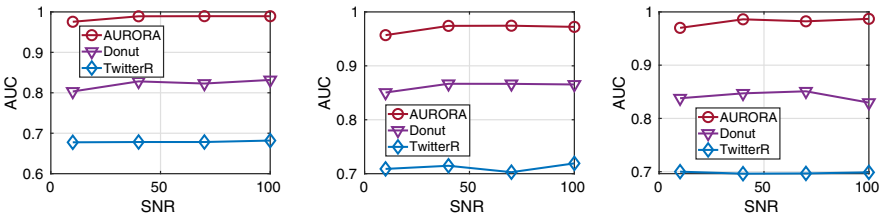
worth noting that more optimistic metrics have been employed where any overlap of a predicted interval with anomalous interval is declared a TP (De Paepe et al. 2019), however, we employ the above time-point-wise metric as it does not leave ambiguity about the correspondence of predicted and GT time points.

To quantify the evaluation of period learning we compare the top-k obtained periods with the ground truth (GT) periods, where k is the number of GT periods. We report the accuracy of period identification for datasets with known GT periods.

6.3 Anomaly detection in synthetic time series

We compare the performance of AURORA and baselines for anomaly detection in three types of settings: point-only, contextual-only, and mixed anomalies (Fig. 2). For this analysis we vary the signal-to-noise ratio (SNR) and report an average AUC of ten samples of data for each setting. With decreasing noise level (or increasing SNR), the average AUC of AURORA increases slightly in all three settings and consistently dominates that of alternatives. In the case of point anomalies, AURORA achieves an AUC of 0.98 at SNR greater than 20, while Donut and TwitterR peak at AUC of 0.83 and 0.68, respectively. AURORA is similarly better than baselines in the cases of contextual anomalies and mixture of anomalies, exhibiting a 15% improvement over Donut and a 25% improvement over TwitterR. AURORA's advantage is due to its explicit modelling of normal periodic and trending behavior in the time series which is built into our synthetic datasets. This allows AURORA to precisely detect time points that deviate from normal.

TwitterR's performance is close to constant at different SNRs as it employs LOESS local smoothing to extract a seasonal component. We parametrize TwitterR with the smallest ground truth period in our synthetic data, and hence, its quality is not affected by incorrect periodicity estimation. However, when this important information is not available (e.g. in real data), its reliance on accurate periodicity estimation becomes a crucial step for TwiterR and a potential weak point, limiting its application. Another key drawback of TwitterR is its assumption of a single period in the data. Our synthetic data features complex/multiple periods, which is another factor for AURORA's edge in performance over TwitterR in this set of experiments.



(a) Point anomalies **(b)** Contextual anom. **(c)** Mixture anomalies

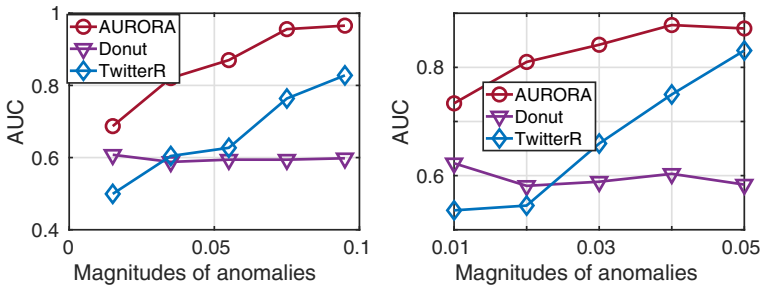
Fig. 2 Comparison of anomaly detection quality for different types of anomalies in synthetic and varying signal-to-noise ratio (SNR) taking values of [10, 40, 70, 100]. We consider **a** point anomalies; **b** contextual anomalies (size range: [3, 4, 5, 6]); **c** mixture of both types

The performance of Donut is also close to constant over varying SNRs. This method uses pre-trained models to score anomalies, i.e., its anomaly score function is bounded by the quality of its training data. Specifically, the VAE at the core of Donut should ideally be trained on anomaly-free data, thus, obtaining a model for normal behavior. The presence of anomalies in the input, however, are incorporated in the model, and thus, affect the likelihood scoring of anomalies in testing data. In contrast, AURORA has no requirement for anomaly-free data. Instead, it models anomalies in the testing data directly without pre-training.

6.4 Anomaly detection in real-world data

We next present anomaly detection results in real-world datasets in Fig. 3. We inject anomalies into the Power plant and Google flu time series. Since difference in performance between point-based and context anomalies is minimal (as observed in synthetic data in Fig. 2), we focus on point anomalies and inject those in our two datasets without GT. The magnitude of anomalies plays an important role as anomalies of increasing magnitude are naturally expected to be more discernible. Hence, we evaluate the performance by varying the magnitude of anomalies.

The AUC of AURORA and TwitterR grows with the magnitude. The AUC of AURORA is greater than 0.9 in most cases and is about 0.3 higher than that of TwitterR in Power plant and 0.25 higher in Google flu. The single period assumption of TwitterR is not necessarily realistic for these datasets and the more flexible periodicity model in AURORA may partially explain its performance advantage. Donut exhibits worse performance in both datasets with injected anomalies and notably its AUC does not grow with the magnitude of injected anomalies. The key reason for this behavior is that Donut requires anomaly-free data for training, but this is often not possible in real-world applications due to the presence of noisy or unidentified anomalies. As a consequence, since we train it on the actual data with injected anomalies to allow for fair comparison, its performance is proportionately affected by the magnitude of anomalies in training data which gets encoded in the VAE. We use part of the actual



(a) Google flu **(b)** Power plant

Fig. 3 Comparison of AUC for anomaly detection by varying the magnitude of injected anomalies in the **a** Google flu; **b** Power plant datasets

data for training the VAE as we do not have access to additional anomaly-free data from the same sources.

The Yahoo and NAB datasets include GT anomaly labels, hence, we present the AUC values for each time series in those benchmarks in Tables 1 and 2. Some time series in the Yahoo benchmark have anomaly labels of questionable quality as reported by others (De Paepe et al. 2019). We mark these time series with unreliable GT labels by an asterisk in Table 1, but show results on all time series for completeness.

AURORA's anomaly detection quality dominates that of baselines in most time series for both benchmarks. In particular, AURORA obtains the best results in 46 out of the 67 time series in the Yahoo benchmark. We get the best results in 29 out of

Table 1 AUC comparison for anomaly detection in the Yahoo benchmark. The performance of the best method in each benchmark dataset are marked in bold font

	1	2	3*	4	5	6	7*	8	9*	10
AURORA	**0.99**	**0.99**	**0.99**	**0.99**	**1**	**1**	**0.79**	**0.98**	**0.99**	**1**
Donut	0.58	0.04	0.02	0.34	0.61	0	0.47	0.42	0.63	0.46
TwitterR	0.52	0.6	0.56	0.42	0.52	0.39	0.59	0.61	0.61	0.75
	11	12	13*	14	15*	16*	17	18*	19	20*
AURORA	**0.99**	**0.96**	**0.86**	**0.73**	**0.84**	**1**	**0.99**	0	**0.99**	**0.58**
Donut	0	0.11	0.3	0.17	0.17	0.41	0.45	0.41	0.77	0.57
TwitterR	0.35	0.35	0.42	0.55	0.46	0.51	0.48	0.56	0.46	0.54
	21	22	23	24	25	26*	27*	28*	29*	30*
AURORA	**0.99**	**1**	0	0	**1**	0	**0.86**	0	0.3	**0.99**
Donut	0.98	0.78	0.49	0.05	0.31	**0.52**	0.5	0.42	0.44	0.03
TwitterR	0.85	0.45	**0.53**	**0.51**	0.50	0.38	0.13	0.52	0.55	0.92
	31	32	33	34*	35*	36*	37*	38*	39*	40
AURORA	**0.91**	0	**0.99**	**0.79**	0	**0.99**	0.43	**0.84**	**0.84**	0.01
Donut	0.71	**0.54**	0	0.72	0	0.38	0.71	0.45	0.47	**0.49**
TwitterR	0.5	0.5	0.25	0.37	0	0.72	**0.70**	0.74	0.66	0.45
	41	42	43	44*	45	46*	47*	48	49*	50
AURORA	**0.98**	**0.99**	**0.58**	0.25	**1**	0.02	0.35	**0.75**	0	**1**
Donut	0.41	0.42	0.64	**0.57**	0.97	0.49	0.42	0.36	0.41	0
TwitterR	0.53	0.47	0.45	0.52	0.17	**0.52**	**0.49**	0.44	**0.83**	0.6
	51*	52*	53	54*	55*	56*	57*	58	59*	60*
AURORA	0.39	**0.64**	0.11	0.5	0.23	0.34	**0.76**	**1**	0	**0.99**
Donut	0.70	0.58	0.1	**0.79**	0.4	0.53	0.53	0.48	0	0.3
TwitterR	**0.67**	0.47	**0.55**	0.3	0.52	**0.81**	0.58	0.48	0	0.52
	61	62	63	64*	65*	66	67			
AURORA	**0.54**	**0.99**	**1**	0	**0.93**	**0.98**	**0.99**			
Donut	0.43	0.44	0	0	0	0.41	0.51			
TwitterR	0.54	0.36	0.12	0	0.48	0.44	0.56			

Indices marked with asterisk (*) have been reported to have questionable anomaly labels in recent work (De Paepe et al. 2020), but we include them for completeness

Table 2 AUC comparison for anomaly detection on the NAB benchmark, realTwitter series. The performance of the best method in each benchmark dataset are marked in bold font

	APPL	AMZN	CRM	CVS	FB	GOOG	IBM	KO	PFE	UPS
AURORA	0.76	**0.99**	**0.99**	**0.55**	**1**	**0.88**	**0.53**	**0.99**	**0.99**	**0.98**
Donut	**0.90**	0.64	0.93	0.42	0.52	0.48	0.49	0.85	0.56	0.77
TwitterR	0.72	0.97	0.94	0.36	0.82	0.59	0.40	0.76	0.73	0.85

34 from the non-questionable labeled time series. In NAB, AURORA gets the best results in 9 out of the 10 time series. Inspecting the cases in which AURORA is not the best method among the baselines reveals that it under-performs in data which does not match well our modeling assumptions of periodicity and trends.

6.5 The importance of multivariate analysis

Recall that we address anomaly detection in multivariate time series. Our approach takes full advantage of shared periods and/or trends among individual univariate time series. Univariate anomaly detection treating each time series as independent cannot take advantage of such shared patterns. To demonstrate the utility of multivariate analysis, we compare AURORA and a univariate version which fits periodicity and trends independently for each time series. To this end, we remove the low-rank regularization term $\lambda_2 \|W\|_*$ from Eq. 9 and call the resulting method AURORAuni. In Fig. 4, we present a comparison between the alternatives on synthetic and real-world data. Both results show that AURORA outperforms AURORAuni with an increase of the AUC of at least 0.1.

6.6 A case study: the Google Flu dataset

Next we employ AURORA on the Google Flu dataset without injecting anomalies) in order to qualitatively explore the reported anomalies in the raw data. Overall, we find that anomalous time points fall well within typical flu seasons.

As an example, we dive deeper into the anomalies detected for Brazil, Bolivia and the United States (Fig. 5). In Brazil, the flu season generally spans May to July, which are the southern hemisphere's winter months (Alonso et al. 2007). Out of the top 50 anomalies detected for Brazil, 44 fall within this time period. The other 6 points all fall in August, with 5 in August 2009 which corresponds with the outbreak of H1N1 flu or swine flue. Brazil registered 7569 new cases of H1N1 flu from August 25 to 29, and later confirmed that the country had the highest number of fatalities from the virus in early September (CNN 2009).

In Bolivia, the flu season typically spans April to September (US Embassy in Bolivia 2019). Out of the top 50 anomalies detected for Bolivia, 47 fall within this time period, and the remaining 3 are in October 2005 and 2007 which might signify a slightly longer flu season in those years.

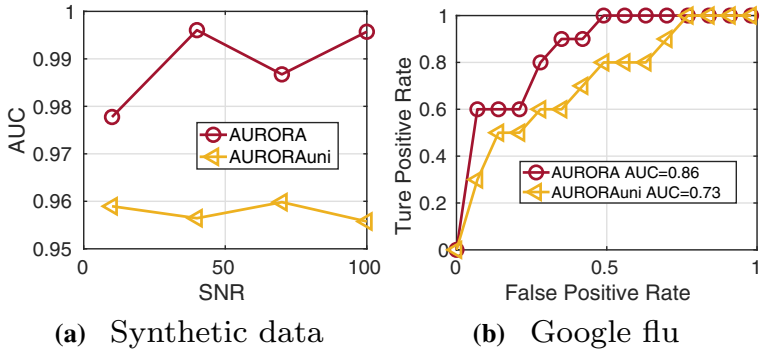


Fig. 4 Comparison between AURORA and a univariate alternative AURORAuni on **a** synthetic and **b** the Google Flu datasets

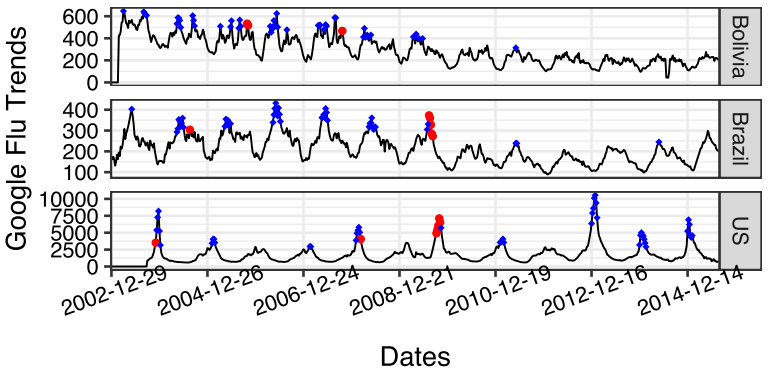
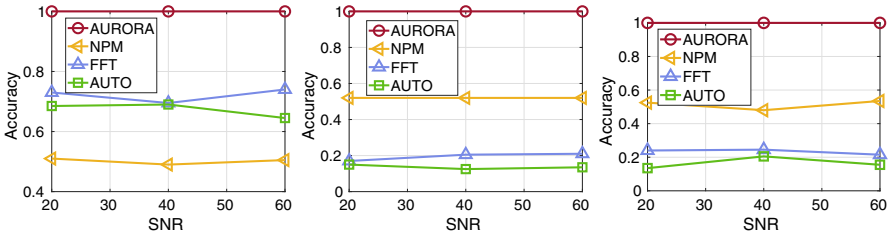


Fig. 5 Case study on Google flu. Blue diamonds and red circles annotate anomalies detected within and outside of flu season, respectively

In the United States, the flu season typically spans fall and winter, and flu activity peaks between December and February (Centers for Disease Control and Prevention 2018). 43 out of our 50 top anomalies detected for the US fall within this time period. Another 6 are in September through November 2009, which coincides with the outbreak of the H1N1 influenza virus that peaked in October (Centers for Disease Control and Prevention 2009). The remaining anomaly is in early March 2008, which indicates a slightly longer flu season, as supported by the fact that the peak flu activity that year was in mid-February.

6.7 Period learning on synthetic data

In Fig. 6, we present AURORA's performance on period learning with varying SNR. AURORA shows superiority over alternatives by consistently obtaining all GT periods. The robustness of AURORA can mostly be attributed to the application of the L1-norm fitting function that is robust to anomalies and noise. In addition, AURORA, detects periods by multivariate analysis and explicitly models trends unlike the alternatives.



(a) Point anomaly **(b)** Contextual anomaly **(c)** Mixture anomaly

Fig. 6 The comparison of period learning by varying SNR under different types of anomalies

NPM achieves 50% accuracy in period detection across settings as it does not model noise or anomalies which may dominate its objective and obs.cure true periods. FFT and AUTO have similar performance as they both employ Fourier Transform. The two methods exhibit different behavior under different types of anomalies. Their accuracy is about 70% for point anomaly and much lower on contextual and mixed (about 20%). This is because point anomalies distort the general periodicity of the time series to a lesser extent than the lengthier contextual anomalies.

6.8 Parameter sensitivity analysis

We study the sensitivity of AURORA to parameters $\{\lambda_1, \lambda_2, \lambda_3\}$ in the synthetic data. We fix one parameter and vary the other two, and report the AUC for anomaly detection in Fig. 7. AURORA achieves a stable close-to-optimal performance for a wide range of parameters. Its minimum AUC in the studied range is around 0.85 which is still better than competitors. The performance of AURORA degrades slightly when the values of parameters approach 1 as large weights on regularization terms undermine the importance of the data fit cost.

6.9 Scalability

We also measure the CPU running time of the three competing anomaly detection methods on synthetic data by varying the length of time series and the number of samples in Fig. 8. We run AURORA on a desktop Dell computer with Intel(R)- Core(i7) CPUs of 3.20 GHz and 31.2 GB memory using MATLAB R2018b 64-bit edition without parallel operation. Donut is executed on the same machine with Python 3.7 without parallel operation. TwitterR is implemented and executed on Intel(R)- Core(i5) CPUs of 2.7 GHz and 8 GB RAM without parallel operation. In a dataset with 0.5 million time points and 200 samples AURORA completes in 14 seconds. In contrast, on the same data size TwitterR requires 150 hours (over 6 days) to run, and Donut requires 1500 hours (over 2 months), including training and testing. Note that since TwitterR and Donut analyze each univariate time series independently, to obtain these results for the largest data size, we ran only a single univariate time series and multiplied the running time by the number of samples: 200. As a result, AURORA is more than 38,000 times faster than TwitterR and more than 380,000 times faster than Donut. Even

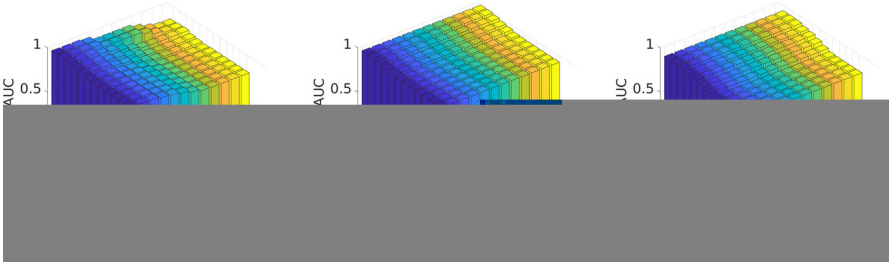


Fig. 7 Parameter sensitivity of AURORA

Fig. 8 Comparison of CPU running time as a function of the time series length and the number of samples (univariate time series) between **a** AURORA, **b** TwitterR and **c** Donut. Note, that AURORA's time is reported in seconds while those of baselines in hours

if we account for differences between programming languages and CPUs, AURORA is still orders of magnitude faster than the baselines.

7 Conclusions

In this paper, we proposed a novel solution for the problem of anomaly detection in multivariate time series in the presence of seasonality and trends. We introduce an efficient and accurate solution, called AURORA, which offers interpretable summaries of the time series and automatic unsupervised anomaly detection. We applied AURORA on both synthetic and real-world datasets and demonstrated its superior performance compared to that of state-of-the-art baselines. AURORA was able to achieve $AUC = 0.95$ for anomaly detection even in very high noise regimes, and 100% accuracy for period learning. In addition, AURORA is orders of magnitude faster than baselines.

Acknowledgements This work is partially supported by the National Science Foundation (NSF) Smart and Connected Communities (SC&C) grant CMMI-1831547. Financial support is also gratefully acknowledged from a Xerox PARC Faculty Research Award, National Science Foundation Awards 1455172, 1934985, 1940124, and 1940276, USAID, and Cornell University Atkinson Center for a Sustainable Future.

References

Alonso WJ, Viboud C, Simonsen L, Hirano EW, Daufenbach LZ, Miller MA (2007) Seasonality of influenza in Brazil: a traveling wave from the Amazon to the subtropics. Am J Epidemiol 165(12):1434–1442

Aminikhanghahi S, Cook DJ (2017) A survey of methods for time series change point detection. Knowled Inform Syst 51(2):339–367

An J, Cho S (2015) Variational autoencoder based anomaly detection using reconstruction probability. Special Lect IE 2(1):1–18

Bleakley K, Vert JP (2011) The group fused lasso for multiple change-point detection. Arxiv preprint arXiv:1106.4199

Boyd S, Parikh N, Chu E, Peleato B, Eckstein J (2011) Distributed optimization and statistical learning via the alternating direction method of multipliers. Found Trends Mach Learn 3(1):1–122

Breunig MM, Kriegel HP, Ng RT, Sander J (2000) Lof: identifying density-based local outliers. SIGMOD Rec 29(2):93–104

Cai JF, Candès EJ, Shen Z (2010) A singular value thresholding algorithm for matrix completion. SIAM J Optim 20(4):1956–1982

Centers for Disease Control and Prevention (2009) Summary of the 2009–2010 influenza season. https://www.cdc.gov/flu/pastseasons/0910season.htm

Centers for Disease Control and Prevention (2018) The flu season. https://www.cdc.gov/flu/about/season/flu-season.htm

Chan PK, Mahoney MV (2005) Modeling multiple time series for anomaly detection. In: Fifth IEEE international conference on data mining (ICDM'05). IEEE

Chen C, Liu LM (1993) Joint estimation of model parameters and outlier effects in time series. J Am Stat Assoc 88(421):284–297

CNN (2009) Brazil says it has most swine flu deaths in world. https://www.cnn.com/2009/WORLD/americas/09/05/brazil.swine.flu/index.html

Davies ME, Plumbley MD (2005) Beat tracking with a two state model [music applications]. In: Proceedings of the IEEE international conference on acoustics, speech, and signal processing (ICASSP'05), vol 3. IEEE, pp iii–241

De Paepe D, Avendano DN, Van Hoecke S (2019) Implications of z-normalization in the matrix profile. In: International conference on pattern recognition applications and methods. Springer, pp 95–118

De Paepe D, Hautte SV, Steenwinckel B, De Turck F, Ongenae F, Janssens O, Van Hoecke S (2020) A generalized matrix profile framework with support for contextual series analysis. Eng Appl Artif Intell 90:

Eilers PHC, Marx BD (2010) Splines, knots, and penalties. WIREs Comput Stat 2(6):637–653. https://doi.org/10.1002/wics.125

Emmott A, Das S, Dietterich T, Fern A, Wong WK (2015) A meta-analysis of the anomaly detection problem. arXiv preprint arXiv:1503.01158

Goepp V, Bouaziz O, Nuel G (2018) Spline regression with automatic knot selection. arXiv preprint arXiv:1808.01770

Goldstein M (2014) Anomaly detection in large datasets. Verlag Dr, Hut

Google (2014) Google flu trends and google dengue trends. https://www.google.org/flutrends

Hautamaki V, Karkkainen I, Franti P (2004) Outlier detection using k-nearest neighbour graph. In: Proceedings of the pattern recognition, 17th International Conference on ICPR'04. IEEE Computer Society, Washington, DC, USA, pp 430–433

Hochenbaum J, Vallis OS, Kejariwal A (2017) Automatic anomaly detection in the cloud via statistical learning. ArXiv arXiv:1704.07706

Hong T, Pinson P, Fan S, Zareipour H, Troccoli A, Hyndman RJ (2016) Probabilistic energy forecasting: global energy forecasting competition 2014 and beyond. Int J Forecast 32(3):896–913

Hundman K, Constantinou V, Laporte C, Colwell I, Soderstrom T (2018) Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery and data mining, KDD '18. ACM, New York, NY, USA, pp 387–395

Indyk P, Koudas N, Muthukrishnan S (2000) Identifying representative trends in massive time series data sets using sketches. In: VLDB, pp 363–372

Jindal T, Giridhar P, Tang LA, Li J, Han J (2013) Spatiotemporal periodical pattern mining in traffic data. In: Proceedings of the 2Nd ACM SIGKDD international workshop on urban computing, UrbComp '13, vol 13. ACM, New York, NY, USA, pp 1–11:8

Keller F, Muller E, Bohm K (2012) HiCS: high contrast subspaces for density-based outlier ranking. In: Proceedings of the 2012 IEEE 28th international conference on data engineering, ICDE '12. IEEE Computer Society, Washington, DC, USA, pp 1037–1048

Killick R, Fearnhead P, Eckley IA (2012) Optimal detection of changepoints with a linear computational cost. J Am Stat Assoc 107(500):1590–1598

Laptev N, Amizadeh S (2015) Yahoo anomaly detection dataset s5. http://webscope.sandbox.yahoo.com/catalog.php?datatype=s&did=70

Laptev N, Amizadeh S, Flint I (2015) Generic and scalable framework for automated time-series anomaly detection. In: Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining, KDD '15. ACM, New York, NY, USA, pp 1939–1947

Lavin A, Ahmad S (2015) Evaluating real-time anomaly detection algorithms—the numenta anomaly benchmark. CoRR arXiv:1510.03336

Li Z, Ding B, Han J, Kays R, Nye P (2010) Mining periodic behaviors for moving objects. In: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, pp 1099–1108

Li Z, Wang J, Han J (2015) ePeriodicity: mining event periodicity from incomplete observations. IEEE Trans Knowl Data Eng 27(5):1219–1232. https://doi.org/10.1109/TKDE.2014.2365801

Lin Z, Chen M, Ma Y (2013) The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. arXiv:1009.5055

Liu FT, Ting KM, Zhou Z (2008) Isolation forest. In: 2008 Eighth IEEE international conference on data mining, pp 413–422. https://doi.org/10.1109/ICDM.2008.17

Liu D, Zhao Y, Xu H, Sun Y, Pei D, Luo J, Jing X, Feng M (2015) Opprentice: towards practical and automatic anomaly detection through machine learning. In: Proceedings of the 2015 internet measurement conference, pp 211–224

Luo X, Nakamura T, Small M (2005) Surrogate test to distinguish between chaotic and pseudoperiodic time series. Phys Rev E Stat Nonlinear Soft Matter Phys 71: https://doi.org/10.1103/PhysRevE.71.026230

Manevitz LM, Yousef M (2001) One-class svms for document classification. J Mach Learn Res 2:139–154

Priestley M (1981) Spectral analysis and time series. Probability and mathematical statistics. Elsevier Academic Press, London (Rep. 2004)

Rosca J, Williard N, Eklund N, Song Z (2015) 2015 PHM data challenge. https://www.phmsociety.org/events/conference/phm/15/data-challenge

Saad EW, Prokhorov DV, Wunsch DC II (1998) Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks. Trans Neur Netw 9(6):1456–1470

Tenneti SV, Vaidyanathan PP (2015) Nested periodic matrices and dictionaries: new signal representations for period estimation. IEEE Trans Sig Process 63(14):3736–3750

US Embassy in Bolivia (2019) Health Alert: U.S. Embassy La Paz, Bolivia. https://bo.usembassy.gov/health-alert-u-s-embassy-la-paz-bolivia-july-15-2019/

Vallis O, Hochenbaum J, Kejariwal A (2014) A novel technique for long-term anomaly detection in the cloud. In: 6th USENIX workshop on hot topics in cloud computing (HotCloud 14)

Van Aken D, Pavlo A, Gordon GJ, Zhang B (2017) Automatic database management system tuning through large-scale machine learning. In: Proceedings of the 2017 ACM international conference on management of data, pp 1009–1024

Vlachos M, Yu P, Castelli V (2005) On periodicity detection and structural periodic similarity. In: Proceedings of the 2005 SIAM international conference on data mining. SIAM, pp 449–460

Wei L, Kumar N, Lolla VN, Keogh EJ, Lonardi S, Ratanamahatana CA (2005) Assumption-free anomaly detection in time series. SSDBM 5:237–242

Xu H, Chen W, Zhao N, Li Z, Bu J, Li Z, Liu Y, Zhao Y, Pei D, Feng Y, Chen J, Wang Z, Qiao H (2018a) Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications. In: Proceedings of the 2018 world wide web conference, WWW '18, pp 187–196

Xu H, Feng Y, Chen J, Wang Z, Qiao H, Chen W, Zhao N, Li Z, Bu J, Li Z, et al (2018b) Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications. In: Proceedings of the 2018 world wide web conference on world wide web—WWW'18

Yeh CM, Zhu Y, Ulanova L, Begum N, Ding Y, Dau HA, Silva DF, Mueen A, Keogh E (2016) Matrix profile i: All pairs similarity joins for time series: a unifying view that includes motifs, discords and shapelets. In: 2016 IEEE 16th international conference on data mining (ICDM), pp 1317–1322

Yuan Q, Zhang W, Zhang C, Geng X, Cong G, Han J (2017) Pred: periodic region detection for mobility modeling of social media users. In: Proceedings of the tenth ACM international conference on web search and data mining, WSDM '17. ACM, New York, NY, USA, pp 263–272. https://doi.org/10.1145/3018661.3018680

Zhang A, Song S, Wang J, Yu P (2017) Time series data cleaning: from anomaly detection to anomaly repairing. Proc VLDB Endow 10:1046–1057. https://doi.org/10.14778/3115404.3115410

Zhang C, Song D, Chen Y, Feng X, Lumezanu C, Cheng W, Ni J, Zong B, Chen H, Chawla NV (2019a) A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. Proc AAAI Conf Artif Intell 33:1409–1416

Zhang L, Bogdanov P (2019) DSL: discriminative subgraph learning via sparse self-representation. In: Proceedings of SIAM international conference on data mining (SDM)

Zhang L, Bogdanov P (2020) Period estimation for incomplete time series. In: IEEE international conference on data science and advanced analytics (DSAA)

Zhang L, Gorovits A, Zhang W, Bogdanov P (2020) Learning period from incomplete multivariate time series. In: IEEE international conference on data mining (ICDM)

Zhang W, James NA, Matteson DS (2017) Pruning and nonparametric multiple change point detection. In: 2017 IEEE international conference on data mining workshops (ICDMW), pp 288–295

Zhang W, Gilbert DE, Matteson DS (2019b) ABACUS: unsupervised multivariate change detection via bayesian source separation. In: Proceedings of the 2019 SIAM international conference on data mining, SDM 2019. Calgary, Alberta, Canada, May 2–4, pp 603–611

Zhou C, Paffenroth RC (2017) Anomaly detection with robust deep autoencoders. In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, pp 665–674

Zhu Y, Shasha D (2002) Statstream: statistical monitoring of thousands of data streams in real time. In: Proceedings of the 28th international conference on very large data bases, VLDB endowment, VLDB '02, pp 358–369