Noname manuscript No.
(will be inserted by the editor)

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

Lin Zhang · Wenyu Zhang · Maxwell J McNeil · Nachuan Chengwang · David S. Matteson · Petko Bogdanov

Received: date / Accepted: date

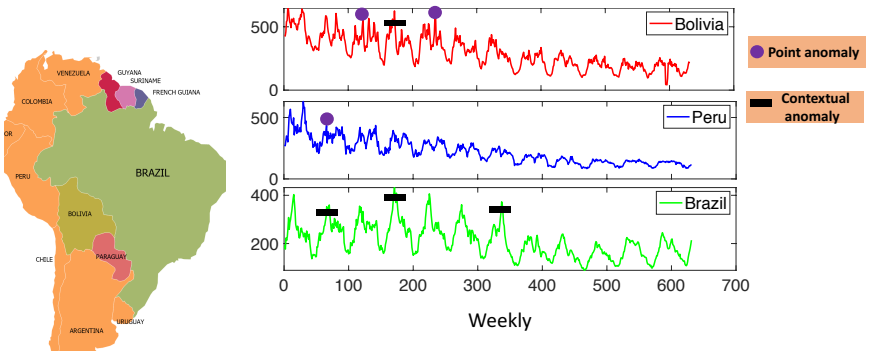
Abstract The ability to accurately and consistently discover anomalies in time series is important in many applications. Fields such as finance (fraud detection), information security (intrusion detection system), healthcare and others all benefit from anomaly detection. Intuitively, anomalies in time series are individual time points or sequence of time points which deviate from normal behavior characterized by periodic oscillations and long-term trends. For example, the 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 which unifies periodicity and long-term trends. To this end, we leverage a Ramanujan periodic dictionary and a spline-based dictionary to capture both seasonal and trending patterns. We conduct experiments on both simulated 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.

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

Lin Zhang, Maxwell J McNeil, Nachuan Chengwang, Petko Bogdanov
University at Albany, SUNY
E-mail: lzhang22, mmcneil2, pbogdanov @albany.edu

Wenyu Zhang, David S. Matteson
Cornell University
E-mail: wz258, matteson@cornell.edu



(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 a weekly record time series of Google flu searches for these three countries spanning from 2002 to 2015.

1 Introduction

Multivariate time series data arise in many domains including the web [49], sensor networks [54], database systems [44], finance [61] and others. Time series from the above domains often exhibit both seasonal behavior and long-term trends [41,38]. For example, city traffic levels [28] and domestic energy consumption [25] 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 [22] over 13 years at one week granularity visualized in Figure. 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, which affect multiple co-evolving times 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 [43], financial fraud loss protection [21], data cleaning [52]. 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 from neighboring time points. 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 need to be detected by making use of long-term contextual information. The detection of anomalies in multivariate time series strengthens our understanding of the behaviour of the whole system. 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 [35,31], 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 unsu-

pervised approaches, including statistical methods [39, 23, 8] and deep-learning [4, 54]. While some existing methods do not rely on explicit anomaly annotations, they have three fundamental limitations: (i) they require normal (non-anomalous) data, (ii) they are sensitive to small local variations, and (iii) they do not offer model interpretability.

In this work, we propose an unsupervised global (or batch) method, called AURORA, to detect anomalies in multivariate time series with an explicit estimation of the normal behavior as 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 follow:
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 on a time series with a half million time points and 200 samples) when compared to state of the art baselines.

2 Related work

Anomaly detection. Existing methods can be broadly categorized into supervised and unsupervised. Supervised anomaly detection methods such as one-class SVM [39] and Isolation Forest [37] employ labeled anomaly data and pose the problem as binary classification [3, 36]. Since labels for anomalies in time series are expensive to obtain and largely unavailable [53], we focus on the unsupervised case. In unsupervised settings, distance-based methods, such as kNN [23], 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 [29]. 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 this structure 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 [11] accounts for temporal correlations and can be extended to include seasonality, but is sensitive to noise and is limited to a single period. TwitterR [24] 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 transform the time series to work in feature space [10], which requires

domain knowledge to construct informative features. In the second category (using comparison across time), by assuming that the historic window is similar to the current window of the time series, the two windows can be compared to identify anomalies [46]. However, it is not straightforward to pick an optimal window size for the assumption to hold. This method also requires some supervision in that it relies on a sanitized (anomaly-free) window of observations.

In view of the wide variety of structures in multivariate time series, recently deep learning techniques have gained increasing attention and have been successfully applied to anomaly detection [53, 26, 4, 48, 60]. In [53], the authors developed a multi-scale model that captures the temporal patterns. Authors of [26] compare LSTM model predictions to actual observations and use thresholding rules to detect anomalies. Several *variational autoencoder (VAE)* methods have also been proposed for this task [4, 48, 60]. The analysis by Zhang and colleagues [53] demonstrates that these deep model methods do not exploit the temporal structure, suffer from noise sensitivity, and do not offer interpretation. In addition, deep learning anomaly detectors also require normal (anomaly-free) data to train their underlying "normal" models. Therefore, these methods are not applicable to fully unsupervised scenarios where normal data is not available. As we demonstrate experimentally, our method consistently outperforms such methods even if they are given access to normal data in both synthetic and real-world applications.

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 more accurate estimation [59], and hence may be limited to detecting contextual anomalies. Some methods require knowledge of the underlying data distributions [30], while others constrain the target change point types, for instance focusing on level shifts [5, 58]. In practice, these methods often fail to correctly detect any anomaly in our settings without additional data treatment since our setting involves data with more complex structure (namely periodicity and trends).

Period learning: Establishing normal periodic behavior requires effective period detection. Traditional period learning work has employed Fourier transform [32, 27] and works in frequency domain to determine pronounced periods. The major drawback of this approach is the detection of a large number of spurious periods [42]. Auto-correlation is another approach for period learning [14] which employs similarity among time segments. These methods rely on a predefined threshold for selecting dominant periods and often employ heuristic post-processing or integration with Fourier spectrograms [45]. Recently, a periodic dictionary based period learning framework has been proposed by Tenneti *et al.* [42], who developed a family of periodic dictionaries and a unified convex model for period detection. Recent work has further improve this by considering the structure in the dictionary [57, 56].

Other recent period learning methods were also proposed for binary sequence data [33, 51]. However, these methods assume that the time series have only one period and rely on prior information about the signals. For example, the model in [33] requires an appropriate segmentation threshold. More importantly, these methods are only applicable to binary sequences and not suited to deal with general

time series. While period detection methods can be employed for estimating normal behavior, they are not directly suited for anomaly detection and do not consider long-term trends.

3 Preliminaries and notation

Notation: We denote by A_i, A^i, A_{ij} the ith column, row of matrix A, and ijth element in 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 ith singular value of A. Also, I denotes identity matrix.

Ramanujan periodic dictionary: Ramanujan periodic dictionary has been proposed by Tenneti *et al.*for learning the underlying periods in univariate time series [42]. For a given period g, the corresponding Ramanujan periodic dictionary is defined as a nested matrix :

$$\Phi_g = [P_{d_1}, P_{d_2}, ...P_{d_K}], \tag{1}$$

where $\{d_1, d_2, ...d_K\}$ 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 [42]:

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

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:

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

To this end, we can obtain the final Ramanujan periodic dictionary R for maximum period g_{max} as $R = [\Phi_1, .., \Phi_{g_{max}}]$, R is also called nested periodic matrix (NPM). To analyze time series of length T, we need to periodically extend the columns in R to the same length as the input time series. Based on this dictionary, we can reconstruct time series with a linear combination of a few columns from R, where the dominant periods of time series correspond to the high-valued learned mixture coefficients.

Spline dictionary: PB-spline regression [20,18] is a flexible method to fit curves using B-splines of degree-d and a smoothness regularization. Quadratic or cubic B-splines are sufficient for most applications [18]. 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 is used to capture 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

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

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 without being sensitive to distortion. The two learned components comprise the normal temporal behavior of the data.

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,

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

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 $\hat{U} = H^{-1}U$.

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 use equally-spaced knots to set 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, which can be formalized as:

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

where W is the coefficient matrix and D is the matrix representation of the difference operator; 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. Intuitively, 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 $1^{st} \sim 3^{rd}$ 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 mentioned 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.

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

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

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 anomalies' locations corresponding to descending magnitude of the residuals. The ordering allows users to analyze each anomaly point according to its prominence. Without any assumption on anomalies' shapes, 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 these two variables alternatively using the Alternating Direction Method of Multiplier (ADMM) framework [7]. We first introduce auxiliary variables: $V = U$, $P = W$ and $Y = X - GU - AW$. Then, we rewrite Eq. 9 as follows:

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

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.

Update Y: The subproblem w.r.t V is written as follows

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

This problem can be solved by the following Lemma from [34].

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

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

which is written as $A_{ij} = sign(B_{ij}) \times max(|B_{ij}| - \alpha, 0)$. 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 of Y:

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

where $E = (X - GU - AW) - \frac{\Gamma_3}{\rho_3}$.
Update V **:** The subproblem w.r.t V is written as below

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

Similarly, we solve this problem using Lemma 1 and get the closed-form solution of V

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

where $H = U - \frac{\Gamma_1}{\rho_1}$.
Update P **:** The subproblem w.r.t P is as follows,

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

According to the singular value thresholding (SVT) method [9], we can compute a closed-form solution for P. 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.
Update U **:** The subproblem w.r.t U is

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

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

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

We get the closed-form solution of U:

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

Update W**:** The subproblem w.r.t W is written as

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

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

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

We obtain the closed-form solution of W:

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

Update Lagrangian multipliers $\Gamma_1 \sim \Gamma_3$: 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 $\Gamma_3^{i+1} = \Gamma_3^i + \rho_3 [Y - (X - GU - AW)]$.

5.1 Overall algorithm and complexity analysis

We summarize AURORA in Algorithm 1. We repeat the sequential 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, like sparse matrix multiplication. For Step 5, the SVD operation has a complexity of $O\left(min\left(tn^2, t^2 n\right)\right)$. 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 incure a cost of $O(q^3)$ and $O(m^3)$, 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 [55]. In the above, S_{nnz} is the number of non-zero elements of $\lambda_3 D^T D + \rho_2 I$ and I_{nnz} is the number of non-zero elements of I. Therefore, the overall assymptotic complexity for each iteration is $O\left(tn^2 + qI_{nnz} + mS_{nnz}\right)$. In particular, AURORA often only needs a few iterations to achieve the convergence, and only cost a few seconds to run in general. *Implementation of our method is available for download at https://www.dropbox.com/sh/obm4tj02zp0zk22/AAB5-XNoMOlNZMgVmTRQhGGVa?dl=0 .*

6 Experimental evaluation

6.1 Datasets

We conduct evaluation experiments on both synthetic and real-world datasets.
Synthetic data: We generate time series data with length 5000 and 20 samples; here, each univariate time series is a sample. These time series include three component: a seasonality component, a trend component and anomalies. We generate the periodic component by following the methodology in [42]. We use polynomial functions to generate the trends in time series. We randomly select individual time points as point anomalies by adding large values to its normal behavior. For each contextual anomalies, we randomly select a segment in time series and add large values. In addition, we add random Gaussian noise to all time series.
Real-world data: We experiment with four real-world datasets, including *Annthyroid* [17], *Mammography* [17], *Power plant* [1] and *Google flu* [22]. The first two datasets have ground truth anomaly labels. For *Power plant* [1] and *Google flu* [22] we manually inject anomalies as ground truth, a common evaluation strategy when anomaly labels are not available [1, 19].

Algorithm 1 AURORA

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

Power plant [1]: This dataset is from the 2015 PHM Society Data Challenge. There are a total of 24 sensors and we extracted a time period of 10000 observations. The sampling interval is 15 minutes. We randomly inject 6 point anomalies for each sensor (144 total), by following our synthetic anomaly injection methodology.
Google flu [22]: The Google flu dataset consists of weekly estimates for influenza rates based off Google searches in 29 countries from 2002 to 2015, including 659 time point per time series. We use this dataset as a case study, and further inject 6 point anomalies to the time series of each country for anomaly detection evaluation.
Annthyroid [17]: This dataset is from the UCI machine learning repository. It contains 7200 observation points and 6 attributes. It has 534 ground truth anomaly time points. The anomalies are Thyroid disease.
Mammography [17]: This dataset is collected by Aleksandar Lazarevic and published at openML. It includes $11,183$ points with 6 attributes, and has 260 labeled anomaly points that are cancer.

6.2 Experimental setup

6.2.1 Baselines

We compare AURORA on anomaly detection with three baselines: TwitterR [24], Donut [49] and Matrix profile [50]. These state-of-the-art anomaly detectors either explicitly account for periodicity and trend, or are flexible in modeling time series patterns. Other anomaly detection methods that we tested are ABACUS [58] and SubspaceCUSUM [47]. These methods have zero accuracy on synthetic data since they assume independently and identically distributed points within time segments, an assumption which does not hold in general for time series analysis. We do not report their accuracy which is consistently close to random.
Donut [49]: This work is based on the Variational Autoencoder (VAE) framework and is the state-of-the-art (SOTA) for anomaly detection in time series. The

anomalies are computed by scoring dependencies in a fix time window based on the trained model. Meanwhile, the periodic and trends are well considered in this model. We report the best result from a wide range window sizes within $[10, 100]$ with a step of 10.

TwitterR [24]: This work introduces the Seasonal Extreme Studentized Deviate (S-ESD), a popular and robust anomaly detection method for univariate seasonal data. Raw data is decomposed into the median, 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 the probability of being an outlier according to the test statistic. The algorithm assumes that the periodicity is known and only allows a single dominant period. In our simulation studies, the period parameter in this method is set to the minimum of the true periods present in each series, so as not to attribute anomaly detection performance to the choice of periodicity learning method. For experiments on real-world data, the true periods are unknown, so the period is chosen according to the frequency with the highest power spectral density in each series.

Matrix profile [50] is a simple yet effective solution for many time series applications including anomaly detection. Working with sliding windows, it computes the minimum distance between a given subsequence and all other subsequences in a time series. Anomalies are declared based on higher than expected measured distances. Intuitively, if an anomaly does not manifest in similar segments elsewhere in the series, it will be successfully flagged.

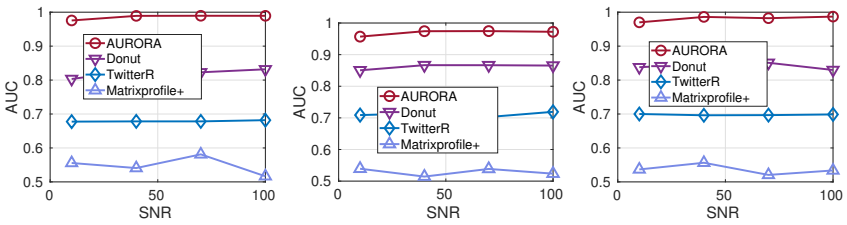
While Matrix Profile does not explicitly handle periodicity and trends, we first remove periodicity and trend components using Seasonal-Trend decomposition via LOESS [12], which is a commonly used method for decomposing time series into periodic and trend components . Next, we execute Matrix profile on the corrected data to detect anomalies detection and denote this version with preprocessing *Matrix profile+*. We experiment with the subsequence length (a major parameter for Matrix Profile) by varying it in the range $[10, 100]$ with a step of 10, and report the best result among these parameter values in the figures.

6.2.2 Performance Metrics

We employ area under the ROC curve (AUC) for evaluating the performance of anomaly detection. For AUC we need to identify true positive classification (TP) vs false positive classification (FP). A TP in the case of point anomalies is the correct identification of the times of the anomalies we injected or ground truth ones. We treat contextual (interval) anomalies as a collection of point anomalies and evaluate AUC for them in the same way.

6.3 Anomaly detection on synthetic data

We evaluate the performance of AURORA on anomaly detection using the three types of anomalies already discussed: Point anomalies, contextual anomalies, and mixture of these two types. We present the results of varying Signal-to-noise ratio (SNR) on different types of anomalies in Fig. 2, where are all comparison results are the averages of ten runs. For decreasing noise level, the AUC of AURORA increases on all types of anomalies and consistently dominates that of alternatives. In the case of point anomalies, AURORA achieves an AUC of 0.98 at SNR greater



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

Fig. 2 The comparison of anomaly detection under different types of anomalies on synthetic data: (a) point anomalies; (b) contextual anomalies; (c) mixture anomalies.

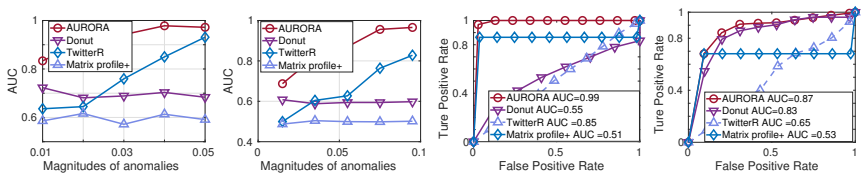
than 20, while Donut and TwitterR only reach up to AUCs of 0.83 and 0.68, respectively. AURORA is similarly better than baselines in the cases of contextual anomalies and mixture anomalies, exhibiting a 15% improvement over Donut and a 25% improvement over TwitterR. The success of AURORA is due to its explicit modelling of the normal periodic and trending behavior in time series. This allows AURORA to precisely detect time points that deviate from normal.

TwitterR's performance is fairly constant at different SNR, because it employs a robust approach where LOESS local smoothing is used to extract the seasonal component. Moreover, we supplied TwitterR with the smallest ground truth period. This means its performance is not affected by periodicity estimation in the synthetic data, but on real data this information is generally not available. This makes periodicity estimation a crucial step for TwiterR and a potential weak point, limiting its application. Another drawback of TwitterR is that the algorithm can only model one periodicity, hence its worse performance compared to AURORA which can model multiple periodicities simultaneously.

The performance of Donut is close to constant due the robustness of scoring anomalies using a trained model. However, the anomaly score function of Donut is biased, and thus lead to an inferior performance to AURORA. More formally, VAE is a generative model, and the training data could be very different from the testing data due to the presence of anomalies. As such, the score function in Donut cannot guarantee to express the true likelihood in testing data. In contrast, AURORA has no assumption on data, and models anomalies from data directly without training.

Matrix profile+ exhibits the worst performance among all methods, close to 0.5 AUC. This is because Matrix profile computes the local similarity by comparing with segments from training time series. Due to the sparsity of anomalies, normal data points dominate the similarity computation and lead to a high index of normality.

Based on Fig. 2, the baselines' performance varies for different types of anomalies, and AURORA has the best performance across all three setups. AURORA models normal behavior based on smoothness and point anomalies "distort" smoothness more significantly than contextual ones. This suggests that between point and contextual anomalies, AURORA has better detection performance for the former. Between point and contextual anomalies, Donut has better performance on the latter, which is about 0.02 higher AUC than its performance for point anomalies. This can be attributed to its analysis of the dependency within a segment such that



(a) Power plant (b) Google flu (c) Annthyroid (d) Mammography

Fig. 3 AUC of anomaly detection by varying the magnitude of anomalies: (a) Power plant; (b) Google flu; c) Annthyroid; d) Mammography

it is more sensitive to contextual anomalies. TwitterR also exhibits better performance for contextual (0.03 higher AUC) as compared to point anomalies, possibly because dispersed point anomalies "throw off" the local regression procedure.

6.4 Anomaly detection in real-world data

The anomaly detection results in the real-world datasets are reported in Fig. 3. Recall that we inject anomalies into the Power plant and Google flu datasets. Since the performance difference among different types of anomalies are not significant in general, we only use point anomaly for injection. Intuitively, the magnitude of anomalies plays an important role in anomaly detection, as a large magnitude is more discernible than a small one. As such, we evaluate the performance by varying the magnitude of anomalies. The performance of AURORA and TwitterR grows quickly when increasing the magnitude. As expected, the AUC of AURORA is greater than 0.9 in most cases, which is about 0.3 higher than TwitterR in Power plant and 0.25 in Google flu. As real-world data is often noisy and not ideally seasonal with only one period, the seasonality assumption in TwitterR is undermined. With a more flexible definition and explicit approximation of the normal time series pattern and a robust fitting function, AURORA is able to detect anomalies efficiently. Donut, however, achieves the worst performance in both datasets and has only small variations in AUC when varying the magnitude. Donut demands anomaly free data for training, but this is often not accessible in real-world application due to the presence of noisy and unidentified anomalies. As a consequence, the performance of Donut is bounded by the training data. Without near perfect training data, the performance of Donut will be weak in general.

Matrix profile+ exhibits poor performance here similar to that in synthetic data. Besides the reasons mentioned in the synthetic data discussion, where the anomalies are sparse, another reason for this performance is the difficulty of finding clean reference time series for Matrix profile + to conduct comparisons in the real-world cases. Real-world data is generally noisy and moreover, since the system undergoes periodic oscillations and long-term trends and hence is not stationary, there is no guarantee that the subsequences from historical data are representative of the normal state of the current system. Note that the pre-processing steps to remove seasonality and trends may introduce extra errors to anomaly detection because the accuracy of pre-processing without taking into account of anomalies is limited.

As ground truth anomaly labels are available in Annthyroid and Mammography, we present the ROC curves of each method. The results show similar behavior

as before. Due to the periodicity and trend modelling on time series, AURORA captures anomalies better than other baselines, where it achieves 0.99 and 0.87 AUC on these two datasets respectively. TwitterR obtains the second best performance in Annthyroid, which is 0.85 AUC. This is possibly due to the presence of a low number of periodicities in the data, which better fit TwitterR's assumption of having a single period. While Donut and matrix profile only obtain 0.51 AUC and 0.53 AUC, respectively. Donut requires the training data to simulate all underlying normal behaviors, this is often difficult due to the lack of prior information of test time series and thus gets a low performance. The low performance of matrix profile is because of it depends on local similarity among fragments, therefore, this may not suitable for detecting anomalous with the presence of long term changes, like periodicity and trends. In the mammography data, AURORA and Donut have similar performance, which are 0.87 AUC and 0.83 AUC, respectively. The reason why Donut gets a good result is because the training data input into Donut is very close to the test data. Therefore, Donut is able to learn the pattern of normal data and distinguish anomalies in the test data. TwitterR and matrix profile achieve much lower AUC than AURORA, which are 0.65 and 0.59 respectively. TwitterR requires single period input data, an assumption which may not be valid for this dataset, which may have complicated patterns, including multiple periods and noise. Matrix profile gets the worst performance because it depends on local similarity, which is easily violated by noise and global changes.

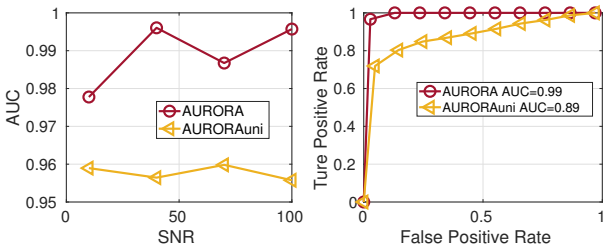
6.5 The importance of multivariate analysis

Recall that we address anomaly detection in multivariate time series. Our approach takes full advantage of the structure among individual univariate time series, including shared trends. Univariate anomaly detection treating each time series as independent will ignore such relations. To demonstrate the necessity of analysing multivariate time series, we compare the proposed method AURORA and its univariate version which fits periodic and trend behavior independetly for each time series. To this end, we remove the low-rank regularization term $\lambda_2 \|W\|_*$ from Eq. 9 and term the resulting method AURORAuni. In Figure. 4, we present a comparison between the alternatives on synthetic and real-world data. Both results show that AURORA outperforms AURORAuni with an AUC of AURORA about 0.1 higher than that of its univariate alternative. The advantage of AURORA highlights the importance of capturing the multivariate structure in time series for anomaly detection.

6.6 Case study

The Google Flu dataset [22] provides Google estimates of influenza activity for 29 countries from 2002-12-29 to 2015-08-09 by aggregating Google Search queries. We apply AURORA on the dataset for anomaly detection, and we find that the anomalies detected correspond well with flu seasons.

As an example, we dive deep into the anomalies detected for Brazil, Bolivia and the United States. In Brazil, the flu season generally spans May to July, which are the southern hemisphere's winter months [2]. Out of the top 50 anomalies detected for Brazil, 44 fall within this time period. The other 6 points all fall in August,



(a) Synthetic data (b) Annthyroid

Fig. 4 Comparison between AURORA and a univariate alternative AURORAuni on (a) Synthetic data and (b) the Annthyroid dataset.

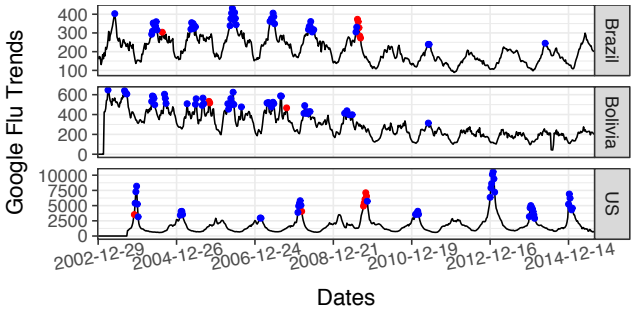


Fig. 5 Case study on Google flu. Blue and red points are anomalies detected within and outside of flu season, respectively.

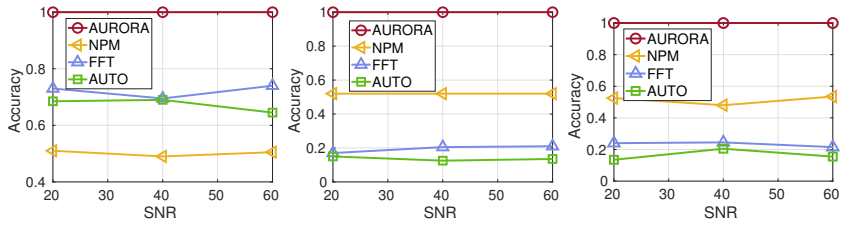
with 5 in August 2009 which corresponds with the outbreak of H1N1 flu or swine flue. Brazil registered 7569 new cases of H1N1 flu from August 25 to 29, and later confirmed that the country had the highest number of fatalities from the virus in early September [13].

In Bolivia, the flu season typically spans April to September [6]. 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.

In the United States, the flu season typically falls in the fall and winter and flu activity peaks between December and February [16]. 43 out of our 50 top anomalies detected for the US fall within this time period. Another 6 are in September to November 2009, which coincides with the outbreak of the H1N1 influenza virus that peaked in October [15]. 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

Baselines: Since AURORA integrates a periodic dictionary, we evaluate the performance of our model on detecting the true periods in raw data and compare its accuracy with the following three baselines. **NPM [42]:** This is the state-of-the-art period learning method based on periodic dictionaries. A time series is reconstructed based on the Ramanujan periodic dictionary. Then, the periods are



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

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

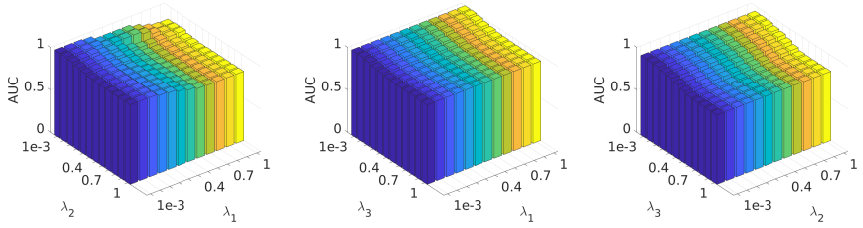
estimated based on the reconstruction coefficients. Since NPM is for univariate time series, we apply it on each univariate time series and add up the top ranked periods as final results. **FFT [40]:** This is a classical period learning method that transforms time series into frequency domain. The periods are determined as the the top ranked coefficients. **AUTO [33]:** This method 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 first step and derives the period of the highest power.

Metric: To quantify the evaluation, we compute the accuracy by comparing the top-k obtained periods with the ground truth (GT) periods, where k is the number of GT periods. Here, we report this evaluation only when the GT periods available.

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 insensitive to anomalies or noisy. NPM only achieves 50% accuracy in general. This is because NPM model considers neither noise nor anomalies. These two abnormalities will dominant the fitting function and undermine the performance of NPM. FFT and AUTO have similar performance as they both depend on Fourier Transform. They show different behavior under different types of anomalies, where they get around 70% accuracy on point anomaly and 20% on other types of anomaly. This is because point anomalies do not break the general shape of a time series, while contextual anomalies corrupt the underlying structure. Therefore, FFT and AUTO obtain better results on the case of point anomaly than others anomalies.

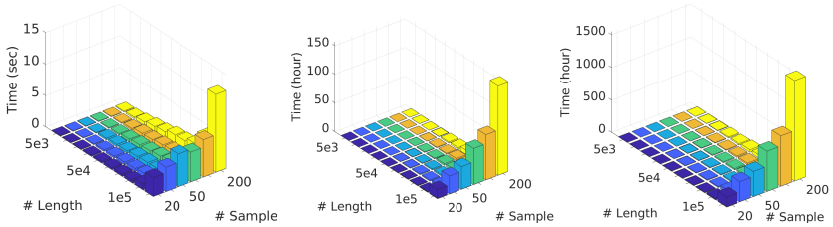
6.8 Parameter sensitivity analysis

In this section, we study the sensitivity of AURORA to parameters $\{\lambda_1, \lambda_2, \lambda_3\}$ on the synthetic data. We fix one parameter and vary the other two, and report the AUC of anomaly detection when varying parameters, as shown in Fig. 7. AURORA is able to achieve a decent performance with parameters in a wide range, which has a minimum AUC around 0.85 and is still much better than competitors. The performance of AURORA slightly degrades when the values of parameters approach close to 1 because large weights on regularization terms will undermine the importance of the cost function, therefore, we should set small values for these three parameters in general.



(a) Varying λ_1 and λ_2 (b) Varying λ_1 and λ_3 (c) Varying λ_2 and λ_3

Fig. 7 Parameter sensitivity of AURORA.



(a) AURORA (**Seconds**) (b) TwitterR (**Hours**) (c) Donut (**Hours**)

Fig. 8 The comparison of CPU running time.

6.9 Scalability

We also measure the CPU running time of three anomaly detection methods on synthetic data by varying the length of time points and the numbder of samples, as shown in Fig. 8. The experiment of AURORA is implemented on a desktop 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 on Intel(R)- Core(i5) CPUs of 2.7 GHz and 8 GB without parallel operation. AURORA is able to execute a data with a half million time points and 200 samples around 14 seconds. In contrast, TwitterR needs around 150 hours to run and Donut consumes around 1500 hours, including training and testing. AURORA is more than 38,000 times faster than TwitterR and more than 380,000 times faster than DONUT. Even if we account for any differences between programming languages and CPUs AURORA is still orders of magnitude faster than its competition.

7 Conclusions and Future work

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. Specifically, we formalize the problem of anomaly detection as minimizing the data reconstruction from the perspective of seasonality and trends. We introduce an efficient solution, called AURORA, offering 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 this work we assumed that the normal behavior of time series consists of a mixture of periodic trend components and formulate anomalies as deviations from these global trends. These two components are usually the most salient features of time series. However, normal behavior in time series may be more complex. For example, it may be the trajectory of a stochastic process with more complex temporal dependencies. The advantages of our model are tied to the periodicity and trend assumptions and further research is necessary to allow for richer normal behavior. In addition, our method requires offline multi-pass analysis of the time series, while many application scenarios demand fast online anomaly detection. Hence, future research is necessary to investigate versions of AURORA in which detection can be performed in an online manner.

References

1. 2015 PHM Data Challenge. https://www.phmsociety.org/events/conference/phm/15/data-challenge (2015). Accessed: 2019-10-26
2. Alonso, W.J., Viboud, C., Simonsen, L., Hirano, E.W., Daufenbach, L.Z., Miller, M.A.: Seasonality of Influenza in Brazil: A Traveling Wave from the Amazon to the Subtropics. American Journal of Epidemiology **165**(12), 1434–1442 (2007)
3. Aminikhanghahi, S., Cook, D.J.: A survey of methods for time series change point detection. Knowl. Inf. Syst. **51**(2), 339–367 (2017). DOI 10.1007/s10115-016-0987-z
4. An, J., Cho, S.: Variational autoencoder based anomaly detection using reconstruction probability (2015)
5. Bleakley, K., Vert, J.P.: The group fused lasso for multiple change-point detection. Arxiv preprint arXiv:1106.4199 (2011)
6. in Bolivia, U.E.: Health alert: U.s. embassy la paz, bolivia (2019). URL https://bo.usembassy.gov/health-alert-u-s-embassy-la-paz-bolivia-july-15-2019/
7. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Found. Trends Mach. Learn. **3**(1), 1–122 (2011). DOI 10.1561/2200000016. URL http://dx.doi.org/10.1561/2200000016
8. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: Lof: Identifying density-based local outliers. SIGMOD Rec. **29**(2), 93–104 (2000). DOI 10.1145/335191.335388
9. Cai, J.F., Candès, E.J., Shen, Z.: A singular value thresholding algorithm for matrix completion. SIAM J. on Optimization **20**(4), 1956–1982 (2010). DOI 10.1137/080738970
10. Chan, P.K., Mahoney, M.V.: Modeling multiple time series for anomaly detection. In: Fifth IEEE International Conference on Data Mining (ICDM'05), pp. 8–pp. IEEE (2005)
11. Chen, C., Liu, L.M.: Joint estimation of model parameters and outlier effects in time series. Journal of the American Statistical Association **88**(421), 284–297 (1993)
12. Cleveland, R.B., Cleveland, W.S., McRae, J.E., Terpenning, I.: Stl: A seasonal-trend decomposition. Journal of official statistics **6**(1), 3–73 (1990)
13. CNN: Brazil says it has most swine flu deaths in world (2009). URL https://www.cnn.com/2009/WORLD/americas/09/05/brazil.swine.flu/index.html
14. Davies, M.E.P., Plumbley, M.D.: Beat tracking with a two state model (2005)
15. for Disease Control, C., Prevention: Summary of the 2009-2010 influenza season (2009). URL https://www.cdc.gov/flu/pastseasons/0910season.htm
16. for Disease Control, C., Prevention: The flu season (2018). URL https://www.cdc.gov/flu/about/season/flu-season.htm
17. Dua, D., Graff, C.: UCI machine learning repository (2017). URL http://archive.ics.uci.edu/ml
18. Eilers, P.H.C., Marx, B.D.: Splines, knots, and penalties. WIREs Comput. Stat. **2**(6), 637–653 (2010). DOI 10.1002/wics.125
19. Emmott, A., Das, S., Dietterich, T.G., Fern, A., Wong, W.K.: A meta-analysis of the anomaly detection problem (2015)

20. Goepp, V., Bouaziz, O., Nuel, G.: Spline regression with automatic knot selection (2018)
21. Goldstein, M.: Anomaly Detection in Large Datasets (2014)
22. Google: Google flu trends and google dengue trends (2014). URL https://www.google.org/flutrends
23. Hautamaki, V., Karkkainen, I., Franti, P.: Outlier detection using k-nearest neighbour graph. In: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 3 - Volume 03, ICPR '04, pp. 430–433. IEEE Computer Society, Washington, DC, USA (2004). DOI 10.1109/ICPR.2004.671
24. Hochenbaum, J., Vallis, O.S., Kejariwal, A.: Automatic anomaly detection in the cloud via statistical learning. ArXiv **abs/1704.07706** (2017)
25. Hong, T., Pinson, P., Fan, S., Zareipour, H., Troccoli, A., Hyndman, R.J.: Probabilistic energy forecasting: Global energy forecasting competition 2014 and beyond. International Journal of Forecasting **32**(3), 896 – 913 (2016). DOI https://doi.org/10.1016/j.ijforecast.2016.02.001
26. Hundman, K., Constantinou, V., Laporte, C., Colwell, I., Soderstrom, T.: Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '18, pp. 387–395. ACM, New York, NY, USA (2018). DOI 10.1145/3219819.3219845
27. Indyk, P., Koudas, N., Muthukrishnan, S.: Identifying representative trends in massive time series data sets using sketches. In: VLDB, pp. 363–372 (2000)
28. Jindal, T., Giridhar, P., Tang, L.A., Li, J., Han, J.: Spatiotemporal periodical pattern mining in traffic data. In: Proceedings of the 2Nd ACM SIGKDD International Workshop on Urban Computing, UrbComp '13, pp. 11:1–11:8. ACM, New York, NY, USA (2013). DOI 10.1145/2505821.2505837
29. Keller, F., Muller, E., Bohm, K.: Hics: High contrast subspaces for density-based outlier ranking. In: Proceedings of the 2012 IEEE 28th International Conference on Data Engineering, ICDE '12, pp. 1037–1048. IEEE Computer Society, Washington, DC, USA (2012). DOI 10.1109/ICDE.2012.88
30. Killick, R., Fearnhead, P., Eckley, I.A.: Optimal detection of changepoints with a linear computational cost. Journal of the American Statistical Association **107**(500), 1590–1598 (2012)
31. Laptev, N., Amizadeh, S., Flint, I.: 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, pp. 1939–1947. ACM, New York, NY, USA (2015). DOI 10.1145/2783258.2788611
32. Li, Z., Ding, B., Han, J., Kays, R., Nye, P.: Mining periodic behaviors for moving objects. In: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 1099–1108. ACM (2010)
33. Li, Z., Wang, J., Han, J.: eperiodicity: Mining event periodicity from incomplete observations. IEEE Trans. Knowl. Data Eng. **27**(5), 1219–1232 (2015). DOI 10.1109/TKDE.2014.2365801. URL https://doi.org/10.1109/TKDE.2014.2365801
34. Lin, Z., Chen, M., Ma, Y.: The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. ArXiv **abs/1009.5055** (2013)
35. Liu, D., Zhao, Y., Xu, H., Sun, Y., Pei, D., Luo, J., Jing, X., Feng, M.: Opprentice: Towards practical and automatic anomaly detection through machine learning. In: Internet Measurement Conference (2015)
36. Liu, D., Zhao, Y., Xu, H., Sun, Y., Pei, D., Luo, J., Jing, X., Feng, M.: Opprentice: Towards practical and automatic anomaly detection through machine learning. In: Proceedings of the 2015 Internet Measurement Conference, IMC '15, pp. 211–224. ACM, New York, NY, USA (2015). DOI 10.1145/2815675.2815679
37. Liu, F.T., Ting, K.M., Zhou, Z.: Isolation forest. In: 2008 Eighth IEEE International Conference on Data Mining, pp. 413–422 (2008). DOI 10.1109/ICDM.2008.17
38. Luo, X., Nakamura, T., Small, M.: Surrogate test to distinguish between chaotic and pseudoperiodic time series. Physical review. E, Statistical, nonlinear, and soft matter physics **71**, 026230 (2005). DOI 10.1103/PhysRevE.71.026230
39. Manevitz, L.M., Yousef, M.: One-class svms for document classification. J. Mach. Learn. Res. **2**, 139–154 (2001)
40. Priestley, M.: Spectral analysis and time series. Probability and mathematical statistics. Elsevier Academic Press, London (1981 (Rep. 2004)). V. 1. Univariate series.– v. 2. Multivariate series, prediction and control.

41. Saad, E.W., Prokhorov, D.V., Wunsch II, D.C.: Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks. Trans. Neur. Netw. **9**(6), 1456–1470 (1998). DOI 10.1109/72.728395
42. Tenneti, S.V., Vaidyanathan, P.P.: Nested periodic matrices and dictionaries: New signal representations for period estimation. IEEE Trans. Signal Processing **63**(14), 3736–3750 (2015)
43. Vallis, O., Hochenbaum, J., Kejariwal, A.: A novel technique for long-term anomaly detection in the cloud. In: HotCloud (2014)
44. Van Aken, D., Pavlo, A., Gordon, G., Zhang, B.: Automatic database management system tuning through large-scale machine learning. pp. 1009–1024 (2017). DOI 10.1145/3035918.3064029
45. Vlachos, M., Yu, P., Castelli, V.: On periodicity detection and structural periodic similarity. In: Proceedings of the 2005 SIAM international conference on data mining, pp. 449–460. SIAM (2005)
46. Wei, L., Kumar, N., Lolla, V.N., Keogh, E.J., Lonardi, S., Ratanamahatana, C.A.: Assumption-free anomaly detection in time series. In: SSDBM, vol. 5, pp. 237–242 (2005)
47. Xie, L., Xie, Y., Moustakides, G.V.: Sequential subspace changepoint detection. arXiv preprint arXiv:1811.03936 (2018)
48. 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.: 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 (2018)
49. Xu, H., Feng, Y., Chen, J., Wang, Z., Qiao, H., Chen, W., Zhao, N., Li, Z., Bu, J., Li, Z., et al.: Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications. Proceedings of the 2018 World Wide Web Conference on World Wide Web - WWW '18 (2018). DOI 10.1145/3178876.3185996
50. Yeh, C.M., Zhu, Y., Ulanova, L., Begum, N., Ding, Y., Dau, H.A., Silva, D.F., Mueen, A., Keogh, E.: 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 (2016)
51. Yuan, Q., Zhang, W., Zhang, C., Geng, X., Cong, G., Han, J.: 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, pp. 263–272. ACM, New York, NY, USA (2017). DOI 10.1145/3018661.3018680. URL http://doi.acm.org/10.1145/3018661.3018680
52. Zhang, A., Song, S., Wang, J., Yu, P.: Time series data cleaning: from anomaly detection to anomaly repairing. Proceedings of the VLDB Endowment **10**, 1046–1057 (2017). DOI 10.14778/3115404.3115410
53. Zhang, C., Song, D., Chen, Y., Feng, X., Lumezanu, C., Cheng, W., Ni, J., Zong, B., Chen, H., Chawla, N.: A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data (2018)
54. Zhang, C., Song, D., Chen, Y., Feng, X., Lumezanu, C., Cheng, W., Ni, J., Zong, B., Chen, H., Chawla, N.: A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data (2019)
55. Zhang, L., Bogdanov, P.: Dsl: Discriminative subgraph learning via sparse self-representation. In: Proceedings of SIAM International Conference on Data Mining (SDM) (2019)
56. Zhang, L., Bogdanov, P.: Period estimation for incomplete time series. In: IEEE International Conference on Data Science and Advanced Analytics (DSAA) (2020)
57. Zhang, L., Gorovits, A., Zhang, W., Bogdanov, P.: Learning period from incomplete multivariate time series. In: IEEE International Conference on Data Mining (ICDM) (2020)
58. Zhang, W., Gilbert, D., Matteson, D.: Abacus: Unsupervised multivariate change detection via bayesian source separation (2018)
59. Zhang, W., James, N.A., Matteson, D.S.: Pruning and nonparametric multiple change point detection. In: 2017 IEEE International Conference on Data Mining Workshops (ICDMW), pp. 288–295 (2017)
60. Zhou, C., Paffenroth, R.C.: Anomaly detection with robust deep autoencoders. In: KDD (2017)
61. Zhu, Y., Shasha, D.: Statstream: Statistical monitoring of thousands of data streams in real time. In: Proceedings of the 28th International Conference on Very Large Data Bases, VLDB '02, pp. 358–369. VLDB Endowment (2002)