
Online Time Series Anomaly Detection with State Space Gaussian Processes

Christian Bock*
ETH Zurich¹

François-Xavier Aubet*
Amazon Research

Jan Gasthaus
Amazon Research

Andrey Kan
Amazon Research

Ming Chen
Amazon Research

Laurent Callot
Amazon Research

Abstract

We propose r-ssGPFA, an unsupervised online anomaly detection model for uni- and multivariate time series building on the efficient state space formulation of Gaussian processes. For high-dimensional time series, we propose an extension of Gaussian process factor analysis to identify the common latent processes of the time series, allowing us to detect anomalies efficiently in an interpretable manner. We gain explainability while speeding up computations by imposing an orthogonality constraint on the mapping from the latent to the observed. Our model’s robustness is improved by using a simple heuristic to skip Kalman updates when encountering anomalous observations. We investigate the behaviour of our model on synthetic data and show on standard benchmark datasets that our method is competitive with state-of-the-art methods while being computationally cheaper.

1 Introduction

Online anomaly detection (AD) in time series data has a wide range of applications, enabling e.g. automatic monitoring & alarming, quality control, and predictive maintenance [25, 39, 17, 12]. One commonly used ansatz to formalise the intuitive notion of an *anomaly* as something that is different from the “normal behavior”, is to define it as an event that is improbable under a probabilistic model of the data. This approach is particularly appealing in the time series setting, where probabilistic models for forecasting are commonly used. With exceptions [18, 17, 8], AD is typically treated as an *unsupervised* machine learning problem, i.e. the training data contains both normal and anomalous instances, but no labels to indicate which is which. In this paper, we follow this paradigm and address the problem of unsupervised time series AD by using a probabilistic time series model based on Gaussian processes (GP).

Recent research in time series AD has focused on improving the detection performance in the large-data regime by leveraging advances in deep learning (DL) [46, 39, 43, 4, 8]. These models can yield high accuracy when sufficient amounts of data are available for training but are not universally applicable. For example, in embedded systems, hardware capabilities or latency requirements may impose severe limitations on model selection. Furthermore, it is often beneficial to encode prior knowledge about the types of *relevant* anomalies [18], a challenging undertaking when data-driven DL models are used. Lastly, more often than not is training data scarce which exacerbates the deployment of data-hungry DL approaches.

Many alternative approaches to deep learning for AD exist and often, heuristics such as thresholds based on statistical moments of the data can be satisfactory. An alternative class of algorithms focuses

*These authors contributed equally

¹Work completed while interning at Amazon Research

on isolating anomalies rather than on constructing models of the *normal* regime [31, 21]. In addition, any time series forecasting model can readily be used for anomaly detection, by declaring points that deviate sufficiently from the predictions as anomalous [9].

In this paper we propose an unsupervised online anomaly detection method for uni- and multivariate time series data that emphasizes computational and sample efficiency, while being flexible enough to model complex temporal patterns and correlation structures. The multivariate variant of our method is based on a factor-analysis variant of multi-output GPs that admit a state space representation for temporal data, enabling efficient linear-time inference. In particular, we make the following contributions:

1. we propose to leverage the state space GP (ssGP) framework [44] to enable linear-time inference in the multivariate time series model Gaussian process factor analysis [51];
2. we propose to enforce an orthogonality constraint on the factor loading matrix, which aids interpretability by decoupling the latent processes and provides an additional speed-up;
3. we propose a simple heuristic for improving the robustness of the Kalman filter inference in the presence of anomalies;
4. we show that our approach can not just detect anomalies, but also provides explainability by attributing them to specific latent components or the noise process.

The remainder of this article is structured as follows: Sec. 2 contextualizes our proposal in the existing literature, followed by a description of the necessary background in Sec. 3. We present our method in Sec. 4, followed by quantitative and qualitative experiments (Sec. 5) and their results (Sec. 6), concluding with a discussion in Sec. 7.

2 Related Work

We first review work on state space models and GPs for time series AD and contrast our proposal with prior art. Then, we detail approaches that focus on adapting GPs to streaming time series and reduce time complexity of GPs.

Early approaches of anomaly detection with *fixed* state state space models go back to the work by Chib and Tiwari [11] where anomalies are detected by a threshold on the predicted noise variance and are followed by statistically motivated methods [45], where the residual of predicted and updated state is treated as a random variable and a statistical hypothesis test is performed to detect anomalies.

GPs have been used in the context of anomaly detection for specific applications such as healthcare monitoring [10, 35]. The closest work to ours is SGP-Q [20], where a sparse variational GP [48] is used and anomaly detection is performed by maintaining a sliding window of the last m observations. After fitting a GP on the window, the probability distribution over the next point is obtained. If this observation exceeds a likelihood threshold under the predictive distribution, it is included in the window, otherwise the mean of the predictive distribution is included. This method differs from ours in four key aspects: 1. Our model allows for faster inference without the need of inverting a (potentially large) kernel matrix. 2. Deciding on the window size parameter is non-trivial and increases memory requirements, which for our model, are solely determined by the (typically small) state space dimension. 3. Our method extends to the multivariate setting and has an inherent explainability component. 4. While we use a similar robustification procedure, ours allows us to treat unlikely points as missing, naturally leading to an increase in the uncertainty of the predictive distribution. These statistical robustification approaches [26] are essential to time series AD and were investigated in the context of Kalman filtering before. They range from elaborate multi-step approaches like the ones by Mu and Yuen [33] or Gandhi and Mili [16] to the approach by Xie and Soh [50] who derive a stable state estimator with bounded error estimates. In the context of anomaly detection Aubet et al. [3] propose a principled Bayesian framework to infer which points of the training set are anomalous, however we prefer to have a robustification procedure directly for the Kalman filtering algorithm.

While the effectiveness of GPs was frequently demonstrated on time series tasks, they do not trivially extend to the streaming setting. Until recently, only little attention had been given to adapting GPs to the streaming setting. Turner [49] proposes a method to drastically speed up inference on equally spaced time series. The linear time complexity sparse variational approximation of GPs [48, 22]

has been adapted to the setting where data arrives sequentially [6]. However, the cost of updating the variational parameter would be prohibitive in a streaming setting. Speeding up the inference in linear multi-output GPs has been proposed using the sparse variational approximation [1, 14], but, to the best of our knowledge, has not been proposed using ssGPs. Lastly, Hou et al. [24] combine the training of conventional GPs with the usage of ssGPs for online camera pose estimation.

3 Background

3.1 Problem statement

Let $\mathbf{y}_{:,1:T} = [\mathbf{y}_{:,1}, \mathbf{y}_{:,2}, \dots, \mathbf{y}_{:,T}]$, denote a multivariate time series, where at each time point t^2 we have an observation vector $\mathbf{y}_{:,t} = [y_{1,t}, y_{2,t}, \dots, y_{D,t}]^T \in \mathbb{R}^D$. Our goal is to decide for each $\mathbf{y}_{:,t}$ whether it is anomalous or not, determining the value of the binary *anomaly indicator* variable $a_t \in \{0, 1\}$. In the online AD setting, observations arrive one at a time, and the anomaly decision for point $\mathbf{y}_{:,t}$ can be based only on observations up to and including that point.

We follow a large body of prior work [41] and approach this problem via a probabilistic model: for each point $\mathbf{y}_{:,t}$ we compute an anomaly score $s_t = -\log p(\mathbf{y}_{:,t} | \mathbf{y}_{:,1:t-1})$ based on the predictive density under a model, and determine a_t by thresholding this score, i.e. $a_t = \mathbb{I}[s_t > \alpha]$ for some fixed decision threshold α .

3.2 State Space Gaussian Processes

A Gaussian process (see [38] for an exhaustive introduction) is a probability distribution over functions for which any finite number of points have a joint Gaussian distribution. A GP is fully specified by its mean function $\mu(\mathbf{x}) : \mathbb{R}^p \rightarrow \mathbb{R}$ and covariance (or kernel) function $k(\mathbf{x}, \mathbf{x}') : \mathbb{R}^p \times \mathbb{R}^p \rightarrow \mathbb{R}$:

$$f(\mathbf{x}) \sim \mathcal{GP}(\mu(\mathbf{x}), k_\theta(\mathbf{x}, \mathbf{x}')). \quad (1)$$

In applications of GPs to time series modeling, the input \mathbf{x} is typically taken to be the time index t , so that the input space dimension is $p = 1$, and the observations are modeled as noisy observations of the function values, i.e. $y_t = f(t) + \varepsilon_t$, with $\varepsilon_t \sim \mathcal{N}(0, \sigma^2)$. The choice of covariance function and its parameters θ determine the properties (e.g. smoothness or periodicity) of the resulting random functions, and can be used to encode prior knowledge into the model.

State space models (SSMs) in general, and linear-Gaussian state space models (LG-SSMs) in particular, are another widely used approach for modeling time series data [15, 40]. A general LG-SSM is defined as follows:

$$\mathbf{y}_{:,t} | \mathbf{z}_{:,t} \sim \mathcal{N}(\mathbf{C}\mathbf{z}_{:,t} + \mathbf{d}, \Psi) \quad \mathbf{z}_{:,t} | \mathbf{z}_{:,t-1} \sim \mathcal{N}(\mathbf{A}\mathbf{z}_{:,t-1}, \mathbf{Q}) \quad (2)$$

where a latent process $\mathbf{z}_{:,t} \in \mathbb{R}^K$ evolves according to linear dynamics parametrized by the *transition matrix* \mathbf{A} driven by Gaussian noise with covariance \mathbf{Q} . The observations $\mathbf{y}_{:,t}$ are a linear function of the latent state plus Gaussian noise with covariance Ψ . Inference (e.g. computing the distribution $p(\mathbf{z}_{:,t} | \mathbf{y}_{:,1:t})$, known as *filtering*) and likelihood computations in this model can be performed in closed form and in linear time (in the number of time steps) via the well-known Kalman filter algorithm [29], and parameters can be learned via maximum likelihood, expectation maximization (EM) [13], or via spectral methods (see e.g. [40, 15] for details).

State Space Gaussian Processes Linear-Gaussian state space models and GPs are closely connected, and in some cases can be seen as two different views on the same underlying model. In particular, for a large class of commonly-used kernel functions, a temporal Gaussian process model can equivalently be expressed in the form of a LG-SSM, where the state transition matrix \mathbf{A} and covariance matrix \mathbf{Q} are derived from the GP kernel and its parameters. The main practical benefit of this conversion is that the model can still be conveniently specified in the form of a covariance kernel, while inference can be performed in linear time using Kalman filtering in the state space representation.

²While our method is applicable to non-equally-spaced time series, we abuse notation and conflate time point and time index here to ease the exposition.

More precisely, for a large class of covariance functions $k(t, t')$ (see Solin [44] for details) a temporal GP model of the form

$$f(t) \sim \mathcal{GP}(\mu(t), k_\theta(t, t')) \quad y_t = f(t) + \varepsilon_t \quad \varepsilon_t \sim \mathcal{N}(0, \sigma^2) \quad (3)$$

can equivalently be expressed as a LG-SSM of the form

$$\mathbf{f}_t = \mathbf{A}_{t-1}\mathbf{f}_{t-1} + \mathbf{q}_{t-1}, \text{ with } \mathbf{q}_{t-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{t-1}) \quad (4)$$

$$y_t = \mathbf{h}^T \mathbf{f}_t + \epsilon_t, \text{ with } \epsilon_t \sim \mathcal{N}(0, \sigma^2) \quad (5)$$

where the state transition matrix \mathbf{A}_{t-1} , the state noise covariance \mathbf{Q}_t , and the emission vector \mathbf{h} are determined by the kernel function $k(t, t')$ and – in the non-equally-spaced setting – depend on the time since the last observation was made. As this is a standard LG-SSM, Kalman filtering and smoothing can be used to obtain the posterior distribution over the latent variable \mathbf{f}_t .

3.3 Gaussian Process Factor Analysis

GPs have been extended to vector-valued functions in the form of multi-output Gaussian processes (MOGPs) (see e.g. [5] and references therein). To avoid computational complexity scaling cubically both in the number of points T and the number of output dimensions D , i.e. $\mathcal{O}(T^3 D^3)$, one can make the assumption that the data can be explained as a linear combination of a small number of latent *factors*. Such *factor analysis* models have a long history and have been explored in the i.i.d. setting [40], in the context of state space models [28], and in the context of GPs [47, 51].

The resulting model, called Gaussian process factor analysis (GPFA) [47, 51], can be described as follows. Characteristically for a factor analysis model, there exists a linear-Gaussian relationship between the D -dimensional observations $\mathbf{y}_{:,t}$, and a K -dimensional latent state $\mathbf{z}_{:,t}$. Each of the K dimensions of the latent state time series is given an independent GP prior, i.e.,

$$\mathbf{y}_{:,t} | \mathbf{z}_{:,t} \sim \mathcal{N}(\mathbf{C} \mathbf{z}_{:,t} + \mathbf{d}, \Psi) \quad (6)$$

$$\mathbf{z}_{k,:} \sim \mathcal{N}(0, K_{\tau\tau}^{(k)}) \quad k = 1, \dots, K \quad (7)$$

where $\mathbf{C} \in \mathbb{R}^{D \times K}$ maps the latent to the observed space with an offset $\mathbf{d} \in \mathbb{R}^{D \times 1}$ and $\Psi \in \mathbb{R}^{D \times D}$ is a diagonal covariance matrix (as in classical factor analysis, each element of its diagonal is the independent noise variance of the corresponding output dimension). $\mathbf{z}_{k,:} \in \mathbb{R}^{1 \times T}$ is the time series corresponding to latent dimension k , and $K_{\tau\tau}^{(k)} \in \mathbb{R}^{T \times T}$ is the covariance matrix of the GP prior on the k^{th} latent dimension, and $\tau = \{1, 2, \dots, T\}$ the set of all the time steps. Each latent process has a different set of parameters for its kernel function.

4 Method

We propose to perform online anomaly detection in multivariate time series using a variant of GP factor analysis [51] combined with the state space GP framework of Solin [44], resulting in a method we refer to as *state space Gaussian process factor analysis* (ssGPFA). Taking advantage of the ssGP framework allows us to efficiently perform (online) inference and learning in the resulting state space model using Kalman filtering in time linear in the number of time steps. To further reduce computational complexity and to gain the ability to explain the anomalies in terms of latent causes, we constrain the columns of the *factor loading matrix* \mathbf{C} to be orthogonal, following ideas proposed by [5] in the context of MOGPs. Finally, we propose a heuristic to improve the robustness of the inference procedure specifically for anomaly detection on streams of data, by foregoing the latent state update for outliers.

4.1 State Space Gaussian Processes Factor Analysis

Our underlying probabilistic time series models is a combination of the GPFA model and ssGPs. In particular, we replace each of the K independent GPs in eq. (7) of the GPFA model with its corresponding ssGP state space model, whose latent state $\mathbf{f}_t^{(k)}$ evolves according to eq. (3). The latent factor $\mathbf{z}_{k,t}$ is then given by $\mathbf{z}_{k,t} = \mathbf{h}^{(k)T} \mathbf{f}_t^{(k)}$.

This allows us to rewrite eq. (6) in terms of the latent state of the corresponding ssGP,

$$\begin{aligned} \mathbf{y}_{:,t} | \mathbf{z}_{:,t} &\sim \mathcal{N} \left(\sum_{k=1}^K \mathbf{C}_{:,k} \mathbf{z}_{k,t} + \mathbf{d}, \Psi \right) \sim \mathcal{N} \left(\sum_{k=1}^K \mathbf{C}_{:,k} \mathbf{h}^{(k)T} \mathbf{f}_t^{(k)} + \mathbf{d}, \Psi \right) \\ &\sim \mathcal{N} \left(\begin{bmatrix} \mathbf{C}_{:,1} \mathbf{h}^{(1)T} & \dots & \mathbf{C}_{:,K} \mathbf{h}^{(K)T} \end{bmatrix} \begin{bmatrix} \mathbf{f}_t^{(1)} \\ \vdots \\ \mathbf{f}_t^{(K)} \end{bmatrix} + \mathbf{d}, \Psi \right) \end{aligned} \quad (8)$$

where $\mathbf{C}_{:,k}$ is the k^{th} column of \mathbf{C} . This reformulation allows us to treat the whole model as one linear state space model in which the state is the concatenation of the states of the K latent ssGPs. The transition and state noise covariance matrices are block diagonal matrices containing respective matrices of the corresponding latent ssGP:

$$\tilde{\mathbf{A}}_t = \begin{bmatrix} \mathbf{A}_t^{(1)} & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{A}_t^{(K)} \end{bmatrix} \quad \tilde{\mathbf{Q}}_t = \begin{bmatrix} \mathbf{Q}_t^{(1)} & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{Q}_t^{(K)} \end{bmatrix}$$

This way, we benefit from all the advantages of ssGPs and GPFA: we can model a multi dimensional time series as lower dimensional processes with different kernel functions and still employ efficient Kalman filtering to use the model on streams of data. We can use the EM algorithm to fit the model parameters as well as the hyperparameters of each of the K kernel functions. In the E-step we use the Kalman filtering and smoothing to obtain the posterior on the latent variables and the M-step is equivalent to the original GPFA model.

4.2 GPFA with Independent Latents

To allow for explainability through disentanglement and to speed up training and inference, one can force the latent processes to be independent of each other a posteriori. We show how this can be achieved in the standard GPFA, here for the inference of the latents for a single time point t , this extends to the inference to the whole time series as we use the E-step formulation presented by Yu et al. [51]. We write the prior on $\mathbf{z}_{:,t}$ as:

$$\mathbf{z}_{:,t} \sim \mathcal{N}(0, \tilde{K}_{tt}) \quad (9)$$

where \tilde{K}_{tt} is a diagonal matrix where the k^{th} entry is the prior variance on $\mathbf{z}_{k,t}$ given by the kernel function evaluated at this point. Using Gaussian conditioning with equations 6 and 9 it follows that the posterior on the latent at time step t is given by:

$$\mathbf{z}_{:,t} | \mathbf{y}_{:,t} \sim \mathcal{N}(\Sigma_t \mathbf{C}^T \Psi^{-1} (\mathbf{y}_{:,t} - \mathbf{d}), \Sigma_t) \quad (10)$$

$$\Sigma_t = \left(\tilde{K}_{tt}^{-1} + \mathbf{C}^T \Psi^{-1} \mathbf{C} \right)^{-1} \quad (11)$$

For the latent to be independent a posteriori, we need Σ_t to be diagonal. As the processes are independent a priori, \tilde{K}_{tt} is diagonal. When a matrix has orthogonal columns, then $\mathbf{C}^T \mathbf{C} = I$. Therefore, the right hand side term is diagonal if one constrains \mathbf{C} to have orthogonal columns and constrains Ψ to be written as $I\sigma^2$.

$$\mathbf{C}^T \Psi^{-1} \mathbf{C} = \mathbf{C}^T I\sigma^2 \mathbf{C} = \mathbf{C}^T \mathbf{C} I\sigma^2 = I\sigma^2 \quad (12)$$

Constraining the columns of \mathbf{C} to be orthogonal means that one cannot simply use closed form update in the M-step (\mathbf{C}^*). At each M-step, we propose to set \mathbf{C} to the closest orthogonal matrix to \mathbf{C}^* in the Frobenius norm. This is also the closest matrix in the KL divergence between the likelihood of the GPFA and the orthogonalised GPFA, since [5] showed that the distance in this KL is proportional to the Frobenius norm between the two matrices. This means that we obtain the orthogonal \mathbf{C} that maximises the free energy at each M-step.

The closest orthogonal \mathbf{C} to \mathbf{C}^* in the Frobenius norm is obtained with singular value decomposition [23]: $\mathbf{C}^* = UDV$, where $U \in \mathbb{R}^{N \times K}$ and $V \in \mathbb{R}^{K \times K}$ have orthogonal columns, and $D \in \mathbb{R}^{K \times K}$

Algorithm 1: Robust Kalman filtering

```

Input: Time series  $\mathbf{t} = (\mathbf{y}_t)_{t=1}^T$ , update threshold  $\rho$ 
1  $o := 1$  // Index last relevant observation
2 for  $t \in \{1, \dots, T\}$  do
3    $\Delta_t \leftarrow t - o$ 
4    $\mathbf{A}_t = \exp(\mathbf{F}\Delta_t)$ 
5    $\mathbf{Q}_t = \mathbf{P}_\infty - \mathbf{A}_t \mathbf{P}_\infty \mathbf{A}_t^T$ 
     // Kalman prediction
6    $\mathbf{m}_{t|t-1}, \mathbf{P}_{t|t-1} \leftarrow$  prediction with Eq. 4
7    $\log\{p(\mathbf{y}_t)\} \leftarrow$  likelihood of observed with Eq. 5
     // Robustify
8   if  $\log p(\mathbf{y}_t) > \rho$  then
9      $\mathbf{m}_{\text{lt}}, \mathbf{P}_{\text{lt}} \leftarrow$  filtering mean and covariance
10     $o \leftarrow t$  // set o to last normal index
11 end

```

is diagonal. We obtain $\mathbf{C} = UV^T$ which has orthogonal columns. This saves us any gradient optimisation to update the model parameters. This procedure is identical to the proposed post-processing step of the original GPFA, with the difference that we do it at every M-step which allows the learned model to give this interpretability.

This constraint allows us to fit each of the latent processes in the E-step in parallel yielding substantial speed gains. In this case, we have a combination of K state space models for which the state and the state transition matrix are the same as the ones of the corresponding latent GP, we only adjust the emission matrix to map to the observed dimension through the column of \mathbf{C} .

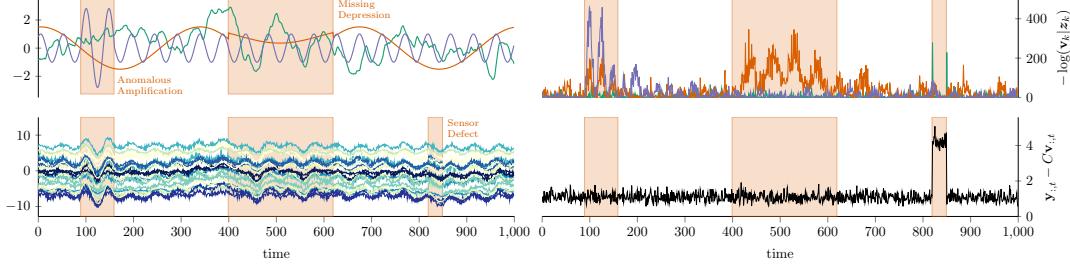
Note that, while the likelihood of FA is not affected by the orthogonalisation of the columns of the loading matrix (proof in Sec. A.1), this is not the case in GPFA, where the likelihood is given by: $p(\mathbf{y}_{:,t}|\theta) = \mathcal{N}(\mathbf{y}_{:,t}|\mathbf{d}, \Psi + \mathbf{C}\tilde{\mathbf{K}}_{tt}\mathbf{C}^T)$. We see that orthogonalising the columns of \mathbf{C} does change the likelihood. In particular, if the unconstrained maximum likelihood loading matrix \mathbf{C} does not have orthogonal columns, the orthogonalisation step will lead to a decrease in likelihood. We argue that in practice, other assumptions tied to the GPFA model, like the linear mapping from the latent to the observed, may also not hold in the true generating process, and so that the additional assumption of having independent latent processes may be acceptable in practice.

4.3 Interpretability and Explainability of Detected Anomalies

ssGPFA allows us to detect anomalies in individual dimensions of a multivariate time series (through the marginal on each dimension) as well as in the whole sequence (through the joint distribution). When constraining the latents to be orthogonal, we can attribute the contribution of each latent dimension to the final anomaly score. In doing so, we can determine “the origin” of an anomaly and provide an interpretation if the latent processes can themselves be interpreted (e.g. the period parameter in Periodic kernels).

Given an anomaly occurring at time t , we rely on the orthogonal columns of \mathbf{C} to obtain the latent values $\mathbf{v}_{:,t}$ that would have best explained the observed point. These values are the ones that minimise $\mathbf{y}_{:,t} - \mathbf{C}\mathbf{v}_{:,t}$ i.e. $\mathbf{v}_{:,t} = (\mathbf{C}\mathbf{C}^T)^{-1}\mathbf{C}^T\mathbf{y}_{:,t}$. The least squared solution is unique when the columns of \mathbf{C} are orthogonal as it results in the dimensions of $\mathbf{v}_{:,t}$ to be independent of each other.

We can compute the likelihood of $\mathbf{v}_{k,t}$ under the probability distribution on $\mathbf{z}_{k,t}$, the corresponding latent process distribution. This likelihood quantifies the contribution of the latent to a point having been classified as anomalous. It may also happen that detected anomaly cannot be explained by the sub-space of the observed space spanned by the columns of \mathbf{C} . In this case, the reconstruction error $\mathbf{y}_{:,t} - \mathbf{C}\mathbf{v}_{:,t}$ will be high. This indicates that we cannot rely on our latent processes to explain the cause of the anomaly (see Sec. 5.1 for an illustration).



(a) Generating latent processes & observed time series (b) Neg. log-likelihoods per latent ssGP & proj. error

Figure 1: Left: three generating latent processes (top) and observed time series (bottom) with anomalous regions. Right: negative log likelihood of the projected latent for each dimension (top) and projection error (bottom).

4.4 Specificities to Online Anomaly Detection

We define the anomaly score of an observation as its negative log-likelihood under the predictive distribution generated for that point which makes each score interpretable.

Robust Kalman Filter State space GPs are commonly used for regression or forecasting tasks where one can assume that all observations are normal. In the AD setting, some points may also be anomalous. We propose a simple and intuitive heuristic to address this problem. If a point is too unlikely, it is not used for the state update, and therefore treated as missing. We introduce a user-defined update threshold ρ , controlling how likely, given the current model, an observation must be to contribute to the update (see lines 7 & 8 of Algorithm 1). The advantages of this method over the ones introduced in Sec. 2 lie in its simplicity and that it leads to an increase in the predictive uncertainty after ignored points.

Computational and memory complexity The computational complexity for **training** with orthogonalisation constraint on \mathbf{C} is $\mathcal{O}(TD^3KL^2 + TKL^3)$, and $\mathcal{O}(TD^3K^2L^2 + TK^3L^3)$ without constraints. L is the maximum state space dimension among all latent processes. For **Inference**, the complexity is $\mathcal{O}(D^3KL^2 + KL^3)$ if \mathbf{C} is constrained, and $\mathcal{O}(D^3K^2L^2 + K^3L^3)$ otherwise (see section B of the appendix for more details). Beyond making the complexity scale linearly in the number of latent dimensions, the orthogonalisation of C allows to compute the filtering, smoothing, and the E-step for each latent process independently in parallel.

5 Experiments

5.1 Explainable Anomaly Detection

We illustrate the explainability capabilities of our model on a synthetic dataset. For this, we first draw three latent time series: one from a Matérn kernel and two from Cosine kernels whose varying periods simulate daily and weekly fluctuations. Anomalies in the *data generating process* are injected by increasing and dampening the short and long period Cosine latents, respectively. Furthermore, a measurement offset is added to 7 of the *observed* dimensions simulating a defect sensor. We train ssGPFA on time series generated from the same underlying yet uncorrupted process, and Figure 1 depicts the results of running the streaming anomaly detection on corrupted *test* sequences. The generating latent processes and observed time series are shown at the top and bottom of Figure 1a.

Figure 1b depicts how the likelihoods of projected latents v_k under the predictive distributions of the model’s latent process (top-right) allows to accurately identify the latents that caused the observed point to be classified as anomalous: The projection of the short period latent has a low likelihood in the first anomaly; the projection of the long period latent is unlikely in the second anomalous region. The third anomaly is an irregularity in the coordination of the observed dimensions with each other. As a result, the observed points are not part of the sub-space that is spanned by the columns of \mathbf{C} in the observed space, which causes a high projection error (bottom-right).

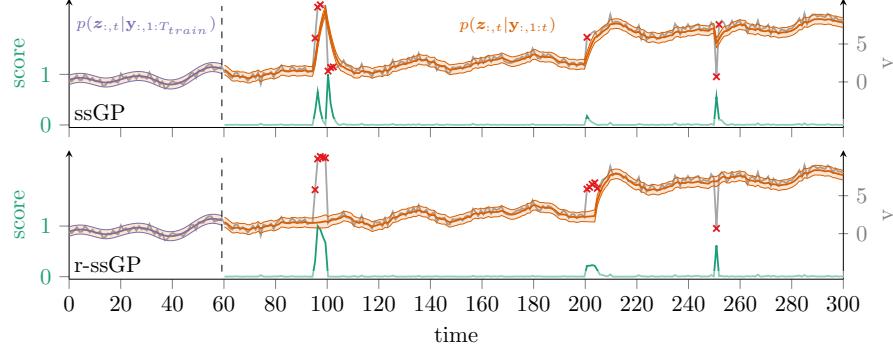


Figure 2: Robust (r-ssGP) and non-robust (ssGP) models on a synthetic time series. Filter distributions are shown in orange, scaled negative log-likelihoods in green. Scores below 0.1 are marked with red crosses.

5.2 Robustness to anomalies

A demonstration of the effect of the proposed robustification of the Kalman update (at $\rho = 1 \times 10^{-12}$) is shown in Figure 2. We generate a synthetic time series with two anomalies and a change point following the equation in C.3. Both robust and non-robust ssGPs are trained on the first 60 data points and their filtering solutions are visualised on the rest of the time series.

The non-robust ssGP rapidly adjusts to anomalies persisting over a longer period (around $t = 100$), resulting in distorted predictions and anomaly scores whereas r-ssGP predictions are unaffected. When a sustained distribution shift occurs (around $t = 200$), the increasing uncertainty of future predictions allows r-ssGP to adjust once the shift can be considered as a new normal regime.

5.3 Detection Performance

Evaluating Anomaly Detectors Label sparsity prevents employing performance metrics such as accuracy. Commonly used metrics are the F_1 -score, precision, and recall redefined for anomaly ranges: Consecutive anomalies are bundled into *one* anomaly range. If a single observation in this range is detected, all observations in this range are considered to be correctly identified. For each method, we perform a search for the best F_1 -score by considering all unique scores a method generated for the data set at hand as decision thresholds. We are well aware of the valid criticism of this evaluation method by Kim et al. [30], however we propose here to demonstrate that our method performs reasonably rather than set a new state-of-the-art result. We think that for our purposes this metric is sufficient.

Data Sets & Preprocessing We use three publicly available data sets for time series anomaly detection to assess detection performance. The multivariate data sets include the NASA data sets (SMAP and MSL) [27] and the Server-Machine Dataset (SMD) [46]. The Numenta Anomaly Benchmark (NAB) [2] is a univariate data set on which we evaluate ssGP, and r-ssGP. SMAP and MSL contain telemetry data from two NASA spacecrafts and NAB, among others, network utilisation and server temperature measurements. SMAP, MSL and SMD are already partitioned in an unlabeled train set and a test set, whereas, NAB has no predefined split. We therefore split each time series into a 20 % training and 80 % testing sequence. For more information about the data sets, we refer to Table 3 in the appendix. We standardise the training sequence to have zero mean and unit variance and use its mean and standard deviation to scale the test sequence accordingly.

Comparison Partners On NAB, we compare to RRCF [21] and HTM. The former is a random cut tree based method commonly used by practitioners and the later achieves state-of-the-art performance on univariate time series. The anomaly scores of HTM are taken directly from their publication to compute its performance. For SMD and NASA, we compare to OmniAnomaly [46], NCAD [8], USAD [4], THOC [43], LSTM-NDT [27], DAGMM [52], and LSTM-VAE [36]. The first four methods represent state-of-the-art neural network approaches. OmniAnomaly combines different ideas from RNNs, variational autoencoders, and normalizing flows, while THOC combines spherical embeddings on multi-scale temporal features with one-class classification. NCAD on the other hand

Table 1: Detection performance on the SMD, MSL, and SMAP benchmark data sets.

| | F_1 | SMD | | MSL | | SMAP | | # Params | Inference time per time point |
|-----------------|-------|-------|-------|-------|-------|-------|-------|----------|-------------------------------|
| | | Prec. | Rec. | F_1 | Prec. | Rec. | F_1 | Prec. | Rec. |
| RRCF | 55.17 | 50.27 | 61.13 | 91.35 | 88.83 | 94.01 | 92.20 | 92.09 | 92.31 |
| OmniAnomaly | 88.57 | 83.34 | 94.49 | 89.89 | 88.67 | 91.17 | 84.34 | 74.16 | 97.76 |
| NCAD | 80.16 | 76.08 | — | 95.47 | 94.81 | 96.16 | 94.45 | 96.24 | 92.73 |
| USAD | 93.82 | 93.14 | 96.17 | 91.09 | 88.10 | 97.86 | 81.86 | 76.97 | 98.31 |
| THOC | — | — | — | 93.67 | — | — | 95.18 | — | — |
| LSTM-NDT | 60.37 | 56.84 | 64.38 | 56.40 | 59.34 | 53.74 | 89.05 | 89.65 | 88.46 |
| DAGMM | 70.94 | 59.51 | 87.82 | 70.07 | 54.12 | 99.34 | 71.05 | 58.45 | 90.58 |
| LSTM-VAE | 78.42 | 79.22 | 70.75 | 67.80 | 52.57 | 95.46 | 72.98 | 85.51 | 63.66 |
| r-ssGPFA (ours) | 73.15 | 67.35 | 80.05 | 73.94 | 60.93 | 94.03 | 71.20 | 97.15 | 56.19 |
| ssGPFA (ours) | 66.93 | 56.50 | 82.07 | 66.05 | 55.64 | 81.26 | 66.54 | 81.15 | 56.39 |

allows to incorporate labels into the one-class classification approach. We take all performance scores from Su et al. [46] and the respective papers of the methods, we run RRCF ourselves.

Implementation and Hyperparameters We implemented our methods using PyTorch [37] and GPy’s [19] SDE kernels as a reference implementation. For all multivariate experiments, we used four versatile Matérn kernels ($\nu = \frac{3}{2}$) with fixed parameters (see Section C.2 for details) For the univariate data set we combined an integrated Brownian motion kernel to cover random fluctuations, a Cosine Kernel to cover periodic behaviour, and a versatile Matérn ($\nu = \frac{3}{2}$). An example of the state space representation of the Matérn kernel is shown in Section C.1, more can be found in [44].

6 Results

Table 1 shows the performance of our model on the *multivariate* data sets. We note the strong positive effect of the robust filtering update on the precision, and overall F_1 -score. This is a particularly desirable property for applications where false positives are very costly. Overall, r-ssGPFA’s performance is competitive with its deep contenders. Interestingly, despite its simplicity, RRCF, a method largely ignored in recent AD works, reaches high performance values on the NASA data sets. The fact that our method with its low number of parameters outperforms a deep method such as DAGMM in all three data sets underlines the effectiveness of our approach. Lastly, our method exhibits the fastest inference time, outperforming RRCF by an order of magnitude.

Table 2: Detection performance on the univariate NAB benchmark data set.

| | F_1 | Prec. | Rec. | # Params | # Hyperparams |
|---------------|-------|-------|-------|----------|---------------|
| RRCF | 82.93 | 76.21 | 90.94 | 0 | 3 |
| HTM | 92.12 | 95.98 | 88.57 | 64k | 17 |
| r-ssGP (ours) | 87.61 | 85.65 | 89.66 | 6 | 1 |
| ssGP (ours) | 90.18 | 90.47 | 89.89 | 6 | 0 |

Table 2 shows the performance assessment for the *univariate* NAB dataset. While ssGP does not beat the state of the art, it outperforms RRCF, the low-parameter comparison partner by a large margin. We observe a performance drop in the robust version of our method, which we attribute to a property of the NAB data set pointed out by Munir et al. [34]: Point-anomalies are centred in an “anomaly window” of observations that are labelled as anomalous while being normal. The range-wise F_1 -score does not take into account as to whether ssGP gets distorted by an anomaly since *normal* observations are labelled abnormal under this anomaly window. Lastly, we compared to SGP-Q [20], a conceptually similar method, on all time series for which the authors provide F_1 -scores: Our method outperforms SGP-Q on 7/9 time series with a median F_1 gain of 3.60.

7 Discussion & Limitations

We present ssGPFA, a method for interpretable and explainable multivariate time series anomaly detection. Combining the advantages of both GPFA and ssGPs, our method exhibits excellent time and space complexity while maintaining competitive detection performance which makes it an attractive option for AD on edge devices. As such, it is particularly attractive for applications in

developing countries where access to large compute clusters may be limited. The GP backbone also allows to generate text descriptions to explain detected anomalies in a similar way as done by the automated statistician [32] for time series characteristics. As with all GP based approaches, however, kernel selection is one of the main disadvantages when prior knowledge about the data is limited. Furthermore, a study of the influence of parameter initialisation and performance consistency with respect to the robustness parameter ρ could shed more light on the inner workings of our method. We aimed to provide a fast and competitive alternative for AD and thus did not investigate this yet.

References

- [1] Vincent Adam, James Hensman, and Maneesh Sahani. Scalable transformed additive signal decomposition by non-conjugate Gaussian process inference. In Francesco A. N. Palmieri, Aurelio Uncini, Kostas I. Diamantaras, and Jan Larsen, editors, *26th IEEE International Workshop on Machine Learning for Signal Processing, MLSP 2016, Vietri sul Mare, Salerno, Italy, September 13-16, 2016*, pages 1–6. IEEE, 2016. URL <https://doi.org/10.1109/MLSP.2016.7738855>.
- [2] Subutai Ahmad, Alexander Lavin, Scott Purdy, and Zuha Agha. Unsupervised real-time anomaly detection for streaming data. *Neurocomputing*, 262:134–147, 2017. URL <https://doi.org/10.1016/j.neucom.2017.04.070>.
- [3] François-Xavier Aubet, Daniel Zügner, and Jan Gasthaus. Monte carlo em for deep time series anomaly detection. *ICML 2021 Time Series Workshop*, 2021. URL [arXiv:2112.14436](https://arxiv.org/abs/2112.14436).
- [4] Julien Audibert, Pietro Michiardi, Frédéric Guyard, Sébastien Marti, and Maria A Zuluaga. Usad: Unsupervised anomaly detection on multivariate time series. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3395–3404, 2020.
- [5] Wessel Bruinsma, Eric Perim, William Tebbutt, J. Scott Hosking, Arno Solin, and Richard E. Turner. Scalable exact inference in multi-output Gaussian processes. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1190–1201. PMLR, 2020. URL <http://proceedings.mlr.press/v119/bruinsma20a.html>.
- [6] Thang D. Bui, Cuong V. Nguyen, and Richard E. Turner. Streaming sparse Gaussian process approximations. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 3299–3307, 2017. URL <https://proceedings.neurips.cc/paper/2017/hash/f31b20466ae89669f9741e047487eb37-Abstract.html>.
- [7] Richard H. Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. A limited memory algorithm for bound constrained optimization. *SIAM J. Sci. Comput.*, 16(5):1190–1208, 1995. URL <https://doi.org/10.1137/0916069>.
- [8] Chris U Carmona, François-Xavier Aubet, Valentin Flunkert, and Jan Gasthaus. Neural contextual anomaly detection for time series. *arXiv preprint arXiv:2107.07702*, 2021.
- [9] Sayan Chakraborty, Smit Shah, Kiumars Soltani, Anna Swigart, Luyao Yang, and Kyle Buckingham. Building an automated and self-aware anomaly detection system. In Xintao Wu, Chris Jermaine, Li Xiong, Xiaohua Hu, Olivera Kotevska, Siyuan Lu, Weija Xu, Srinivas Aluru, Chengxiang Zhai, Eyhab Al-Masri, Zhiyuan Chen, and Jeff Saltz, editors, *IEEE International Conference on Big Data, Big Data 2020, Atlanta, GA, USA, December 10-13, 2020*, pages 1465–1475. IEEE, 2020. URL <https://doi.org/10.1109/BigData50022.2020.9378177>.
- [10] Varun Chandola and Ranga Raju Vatsavai. A Gaussian process based online change detection algorithm for monitoring periodic time series. In *Proceedings of the Eleventh SIAM International Conference on Data Mining, SDM 2011, April 28-30, 2011, Mesa, Arizona, USA*, pages 95–106. SIAM / Omnipress, 2011. URL <https://doi.org/10.1137/1.9781611972818.9>.
- [11] Siddhartha Chib and Ram C Tiwari. Outlier detection in the state space model. *Statistics & Probability Letters*, 20(2):143–148, 1994.
- [12] Andrew A. Cook, Goksel Misirli, and Zhong Fan. Anomaly detection for iot time-series data: A survey. *IEEE Internet Things J.*, 7(7):6481–6494, 2020. URL <https://doi.org/10.1109/JIOT.2019.2958185>.
- [13] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via theEMAlgorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, September 1977. URL <https://doi.org/10.1111/j.2517-6161.1977.tb01600.x>.
- [14] Lea Duncker and Maneesh Sahani. Temporal alignment and latent Gaussian process factor inference in population spike trains. In Samy Bengio, Hanna M. Wallach, Hugo

- Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 10466–10476, 2018. URL <https://proceedings.neurips.cc/paper/2018/hash/d1ff1ec86b62cd5f3903ff19c3a326b2-Abstract.html>.
- [15] James Durbin and Siem Jan Koopman. *Time series analysis by state space methods*. Oxford University Press, 2 edition, 2012.
 - [16] Mital A. Gandhi and Lamine Mili. Robust Kalman filter based on a generalized maximum-likelihood-type estimator. *IEEE Trans. Signal Process.*, 58(5):2509–2520, 2010. URL <https://doi.org/10.1109/TSP.2009.2039731>.
 - [17] Jingkun Gao, Xiaomin Song, Qingsong Wen, Pichao Wang, Liang Sun, and Huan Xu. Robusttad: Robust time series anomaly detection via decomposition and convolutional neural networks. *CoRR*, abs/2002.09545, 2020. URL <https://arxiv.org/abs/2002.09545>.
 - [18] Nico Görnitz, Marius Kloft, Konrad Rieck, and Ulf Brefeld. Toward supervised anomaly detection. *J. Artif. Intell. Res.*, 46:235–262, 2013. URL <https://doi.org/10.1613/jair.3623>.
 - [19] GPy. GPy: A Gaussian process framework in python. <http://github.com/SheffieldML/GPy>, since 2012.
 - [20] Minghao Gu, Jingjing Fei, and Shiliang Sun. Online anomaly detection with sparse Gaussian processes. *Neurocomputing*, 403:383–399, 2020. URL <https://doi.org/10.1016/j.neucom.2020.04.077>.
 - [21] Sudipto Guha, Nina Mishra, Gourav Roy, and Okke Schrijvers. Robust random cut forest based anomaly detection on streams. In Maria-Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 2712–2721. JMLR.org, 2016. URL <http://proceedings.mlr.press/v48/guha16.html>.
 - [22] James Hensman, Nicoló Fusi, and Neil D. Lawrence. Gaussian processes for big data. In Ann Nicholson and Padhraic Smyth, editors, *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence, UAI 2013, Bellevue, WA, USA, August 11-15, 2013*. AUAI Press, 2013.
 - [23] Nicholas J Higham. Matrix nearness problems and applications. In *Applications of Matrix Theory*, pages 1–27. Oxford University Press, 1989.
 - [24] Yuxin Hou, Juho Kannala, and Arno Solin. Multi-view stereo by temporal nonparametric fusion. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 2651–2660. IEEE, 2019. URL <https://doi.org/10.1109/ICCV.2019.00274>.
 - [25] Ruei-Jie Hsieh, Jerry Chou, and Chih-Hsiang Ho. Unsupervised online anomaly detection on multivariate sensing time series data for smart manufacturing. In *12th IEEE Conference on Service-Oriented Computing and Applications, SOCA 2019, Kaohsiung, Taiwan, November 18-21, 2019*, pages 90–97. IEEE, 2019. URL <https://doi.org/10.1109/SOCA.2019.00021>.
 - [26] Peter J Huber. *Robust statistics*, volume 523. John Wiley & Sons, 2004.
 - [27] Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Söderström. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In Yike Guo and Faisal Farooq, editors, *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, pages 387–395. ACM, 2018. URL <https://doi.org/10.1145/3219819.3219845>.
 - [28] Borus Jungbacker and Siem Jan Koopman. Likelihood-based analysis for dynamic factor models. Tinbergen Institute Discussion Paper 08-007/4, 2008. URL <http://hdl.handle.net/10419/86765>.
 - [29] R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1):35–45, 03 1960. ISSN 0021-9223. URL <https://doi.org/10.1115/1.3662552>.

- [30] Siwon Kim, Kukjin Choi, Hyun-Soo Choi, Byunghan Lee, and Sungroh Yoon. Towards a rigorous evaluation of time-series anomaly detection. *arXiv preprint arXiv:2109.05257*, 2021.
- [31] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation-based anomaly detection. *ACM Trans. Knowl. Discov. Data*, 6(1):3:1–3:39, 2012. URL <https://doi.org/10.1145/2133360.2133363>.
- [32] James Robert Lloyd, David Duvenaud, Roger B. Grosse, Joshua B. Tenenbaum, and Zoubin Ghahramani. Automatic construction and natural-language description of nonparametric regression models. In Carla E. Brodley and Peter Stone, editors, *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada*, pages 1242–1250. AAAI Press, 2014. URL <http://www.aaai.org/ocs/index.php/AAAI/AAAI14/paper/view/8240>.
- [33] He-Qing Mu and Ka-Veng Yuen. Novel outlier-resistant extended Kalman filter for robust online structural identification. *Journal of Engineering Mechanics*, 141(1):04014100, 2015. URL <https://ascelibrary.org/doi/abs/10.1061/%28ASCE%29EM.1943-7889.0000810>.
- [34] Mohsin Munir, Shoaib Ahmed Siddiqui, Andreas Dengel, and Sheraz Ahmed. Deepant: A deep learning approach for unsupervised anomaly detection in time series. *IEEE Access*, 7:1991–2005, 2019. URL <https://doi.org/10.1109/ACCESS.2018.2886457>.
- [35] Jingyue Pang, Datong Liu, Haitao Liao, Yu Peng, and Xiyuan Peng. Anomaly detection based on data stream monitoring and prediction with improved Gaussian process regression algorithm. In *2014 International Conference on Prognostics and Health Management*, pages 1–7, 2014. doi: 10.1109/ICPHM.2014.7036394.
- [36] Daehyung Park, Yuuna Hoshi, and Charles C. Kemp. A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder. *CoRR*, abs/1711.00614, 2017. URL <http://arxiv.org/abs/1711.00614>.
- [37] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d’Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8024–8035, 2019. URL <https://proceedings.neurips.cc/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html>.
- [38] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, 2006. ISBN 026218253X. URL <https://www.worldcat.org/oclc/61285753>.
- [39] Hansheng Ren, Bixiong Xu, Yujing Wang, Chao Yi, Congrui Huang, Xiaoyu Kou, Tony Xing, Mao Yang, Jie Tong, and Qi Zhang. Time-series anomaly detection service at microsoft. In Ankur Teredesai, Vipin Kumar, Ying Li, Rómer Rosales, Evinaria Terzi, and George Karypis, editors, *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, pages 3009–3017. ACM, 2019. URL <https://doi.org/10.1145/3292500.3330680>.
- [40] Sam T. Roweis and Zoubin Ghahramani. A unifying review of linear Gaussian models. *Neural Comput.*, 11(2):305–345, 1999. URL <https://doi.org/10.1162/089976699300016674>.
- [41] Lukas Ruff, Jacob R. Kauffmann, Robert A. Vandermeulen, Grégoire Montavon, Wojciech Samek, Marius Kloft, Thomas G. Dietterich, and Klaus-Robert Müller. A unifying review of deep and shallow anomaly detection. *Proc. IEEE*, 109(5):756–795, 2021. URL <https://doi.org/10.1109/JPROC.2021.3052449>.
- [42] Simo Särkkä and Arno Solin. *Applied Stochastic Differential Equations*. Institute of Mathematical Statistics Textbooks. Cambridge University Press, 2019. doi: 10.1017/9781108186735.
- [43] Lifeng Shen, Zhuocong Li, and James Kwok. Timeseries anomaly detection using temporal hierarchical one-class network. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS*

- 2020, December 6-12, 2020, virtual, 2020. URL <https://proceedings.neurips.cc/paper/2020/hash/97e401a02082021fd24957f852e0e475-Abstract.html>.
- [44] Arno Solin. *Stochastic differential equation methods for spatio-temporal Gaussian process regression*. PhD thesis, Aalto University, 2016.
 - [45] Augustin Soule, Kavé Salamatian, and Nina Taft. Combining filtering and statistical methods for anomaly detection. In *Proceedings of the 5th Internet Measurement Conference, IMC 2005, Berkeley, California, USA, October 19-21, 2005*, pages 331–344. USENIX Association, 2005. URL <http://www.usenix.org/events/imc05/tech/soule.html>.
 - [46] Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan Pei. Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In Ankur Teredesai, Vipin Kumar, Ying Li, Rómer Rosales, Evinaria Terzi, and George Karypis, editors, *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, pages 2828–2837. ACM, 2019. URL <https://doi.org/10.1145/3292500.3330672>.
 - [47] Yee Whye Teh, Matthias W. Seeger, and Michael I. Jordan. Semiparametric latent factor models. In Robert G. Cowell and Zoubin Ghahramani, editors, *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics, AISTATS 2005, Bridgetown, Barbados, January 6-8, 2005*. Society for Artificial Intelligence and Statistics, 2005. URL <http://www.gatsby.ucl.ac.uk/aistats/fullpapers/265.pdf>.
 - [48] Michalis K. Titsias. Variational learning of inducing variables in sparse Gaussian processes. In David A. Van Dyk and Max Welling, editors, *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics, AISTATS 2009, Clearwater Beach, Florida, USA, April 16-18, 2009*, volume 5 of *JMLR Proceedings*, pages 567–574. JMLR.org, 2009. URL <http://proceedings.mlr.press/v5/titsias09a.html>.
 - [49] Ryan Darby Turner. *Gaussian processes for state space models and change point detection*. PhD thesis, University of Cambridge, 2012. URL <https://www.repository.cam.ac.uk/handle/1810/242181>.
 - [50] Lihua Xie and Yeng Chai Soh. Robust Kalman filtering for uncertain systems. *Systems & Control Letters*, 22(2):123–129, 1994. ISSN 0167-6911. URL <https://www.sciencedirect.com/science/article/pii/0167691194901066>.
 - [51] Byron M. Yu, John P. Cunningham, Gopal Santhanam, Stephen I. Ryu, Krishna V. Shenoy, and Maneesh Sahani. Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity. In Daphne Koller, Dale Schuurmans, Yoshua Bengio, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 8-11, 2008*, pages 1881–1888. Curran Associates, Inc., 2008. URL <https://proceedings.neurips.cc/paper/2008/hash/ad972f10e0800b49d76fed33a21f6698-Abstract.html>.
 - [52] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Dae-ki Cho, and Haifeng Chen. Deep autoencoding Gaussian mixture model for unsupervised anomaly detection. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL <https://openreview.net/forum?id=BJJLHbb0->.

A GPFA

A.1 Unidentifiability of C in Factor Analysis

It is known that Factor Analysis is not identifiable: multiple solutions provide the same optimal likelihood. Let Q to be an arbitrary orthogonal rotation matrix of size $K \times K$ satisfying $QQ^T = I$. We can obtain a different loading matrix $\tilde{C} = CQ$ for which the likelihood is given by:

$$p(\mathbf{y}_{:,t}|\theta) = \mathcal{N}(\mathbf{y}_{:,t}|\mathbf{d}, \Psi + \tilde{C}\tilde{C}^T) \quad (13)$$

$$= \mathcal{N}(\mathbf{y}_{:,t}|\mathbf{d}, \Psi + CQQ^TC^T) \quad (14)$$

$$= \mathcal{N}(\mathbf{y}_{:,t}|\mathbf{d}, \Psi + CC^T) \quad (15)$$

A.2 Closed form model parameter updates

Using $q(\mathbf{z}_{:,t}) = \mathcal{N}(\mathbf{z}_{:,t}|\boldsymbol{\mu}_t, \Sigma_t)$, the posterior distribution on the latent at each time step t inferred in the E-step, we obtain the closed form updates for the model parameters:

$$\begin{aligned} C^* &= \left(\sum_{t=1}^T (\mathbf{y}_{:,t} - \mathbf{d})\boldsymbol{\mu}_t^T \right) \left(\sum_{t=1}^T (\Sigma_t + \boldsymbol{\mu}_t\boldsymbol{\mu}_t^T) \right)^{-1} \\ \mathbf{d}^* &= \frac{1}{T} \sum_{t=1}^T (\mathbf{y}_t - C\boldsymbol{\mu}_t) \\ \Psi &= \frac{1}{T} \sum_{t=1}^T ((\mathbf{y}_t - C\boldsymbol{\mu}_t - \mathbf{d})(\mathbf{y}_t - C\boldsymbol{\mu}_t - \mathbf{d})^T + C\Sigma_t C^T) \end{aligned}$$

These are the model parameters that maximise the evidence lower bound in the M-step. Note that the derivations and updates are the same as for Factor Analysis.

B Computational and memory complexity

GPs are known to scale very poorly to big datasets but the combination of introduced methods allows for lightweight training and inference. Recall that D is the number of observed dimensions, K the number of latent processes, and T the number of training time steps. We denote L as the maximum state space dimension among all latent processes.

The computational complexity for **training** is $\mathcal{O}(TD^3KL^2 + TKL^3)$ when we constrain the columns of \mathbf{C} to be orthogonal and $\mathcal{O}(TD^3K^2L^2 + TK^3L^3)$ without constraints. In both cases it is linear in the number of training points. When dealing with very high dimensional time series and few latent processes, one can map the input to a K dimensional space proposed by [5] to allow the training complexity to scale linearly in D but cubically in K .

Inference requires the computation of the likelihood of the observation and the Kalman update. The computations complexity is $\mathcal{O}(D^3KL^2 + KL^3)$ if \mathbf{C} is constrained as orthogonal otherwise $\mathcal{O}(D^3K^2L^2 + K^3L^3)$ without constraints. Beyond making the complexity scale linearly in the number of latent dimensions, the orthogonalisation of C allows to compute the filtering and smoothing (and the E-step) for each of the latent process independently of each other. In such setting, it can be easily parallelized only with only little sequential calculation.

C Experiments

C.1 State-Space Matrices

In this section, we exemplify the state space representation of the Matérn Kernel with $\nu = \frac{3}{2}$. For a more comprehensive enumeration of other covariance functions, we refer the interested reader to Solin [44]. In the following example, \mathbf{F} is the feedback matrix of the corresponding SDE, \mathbf{h}^T its measurement model, \mathbf{P}_∞ the stationary state covariance matrix (i.e. the state stabilises to infinity), and \mathbf{A} the *discrete-time* state transition matrix.

State-space matrices for Matern32-kernel with length-scale l , variance σ^2 and let $\lambda = \frac{\sqrt{3}}{l}$:

$$\mathbf{F} = \begin{bmatrix} 0 & 1 \\ -\lambda^2 & -2\lambda \end{bmatrix}, \quad \mathbf{h}^T = [1 \ 0], \quad \mathbf{P}_\infty = \begin{bmatrix} \sigma^2 & 0 \\ 0 & \lambda^2\sigma^2 \end{bmatrix}, \\ \mathbf{A} = \exp(\mathbf{F}\Delta_t)$$

Furthermore, it was shown [44, 42] that the addition and multiplication of covariance functions in state space can be expressed in terms of block diagonals and Kronecker products (\otimes) and sums (\oplus) as shown below.

Kernel Addition The addition of two kernels K_1 and K_2 in the state space formulation is defined by stacking their respective state-space matrices:

$$\mathbf{F} = \text{blkdiag}(\mathbf{F}_1, \mathbf{F}_2), \quad \mathbf{H} = \text{blkdiag}(\mathbf{H}_1, \mathbf{H}_2), \\ \mathbf{P}_\infty = \text{blkdiag}(\mathbf{P}_{\infty 1}, \mathbf{P}_{\infty 2}), \quad \mathbf{A} = \exp(\mathbf{F}\Delta_t)$$

Kernel Multiplication The multiplication of two kernels K_1 and K_2 in the state space formulation is defined by the Kronecker sum of the respective feedback matrices and the Kronecker product of the remaining state matrices:

$$\mathbf{F} = \mathbf{F}_1 \oplus \mathbf{F}_2, \quad \mathbf{H} = \mathbf{H}_1 \otimes \mathbf{H}_2, \\ \mathbf{P}_\infty = \mathbf{P}_{\infty 1} \otimes \mathbf{P}_{\infty 2}, \quad \mathbf{A} = \exp(\mathbf{F}\Delta_t)$$

C.2 Implementation Details

RRCF The RRCF algorithm represents a predefined number of observations (i.e. shingle size) as a higher-dimensional point. In a streaming scenario each new observation gives rise to a new shingle which is used to update the random cut forest. Using reservoir sampling, the tree can be updated in sublinear time which makes this approach particularly attractive for the online setting. The authors define the collusive displacement score (CODISP), a measure of change in model complexity when a new observation is inserted, as their anomaly score. We run RRCF with 40 trees of size 256 and a shingle size of 4.

As for the selected kernel for ssGP, we do not perform a kernel search, but instead choose a versatile Kernel consisting of the addition of a Brownian motion-kernel with a multiplication of a Matern32 and a Cosine-kernel. For ssGPFA, we use four independent ssGPs with fixed Matern32-kernels with lengthscales (130, 200, 50, 10). That being said, our method can readily be used for sophisticated Kernel search algorithms such as [32].

For all methods, we fix the robustification threshold ρ to be $1e-12$ without performing any parameter search. We use the L-BFGS-B [7] algorithm to fit our models for maximally 20 epochs.

C.3 Synthetic Data

The data generating process yielding the time series of Figure 2 is defined by the following equation.

$$f(t) = \cos(0.04t + 0.33\pi) \cdot \sin(0.2t) + \epsilon_{\text{synth.}} + 5/300t, \quad (16)$$

with $\epsilon_{\text{synth.}} \sim \mathcal{N}(0, 0.15)$.

D Data Sets

Table 3 shows data statistics on the used benchmark data sets. \bar{n} refers to the mean time series length.

Table 3: Statistics for the used benchmark data sets. The fraction of anomalies refers to point anomalies, not to anomaly ranges.

| Data Set | Sub Data Set | Num. TS | D | \bar{n} (train)/test | % Anomalies |
|----------|-----------------------|---------|-----|------------------------|-------------|
| NAB | realAWSCloudwatch | 17 | 1 | 3984 | 9.3% |
| | realAdExchange | 6 | 1 | 1601 | 10.0% |
| | realKnownCause | 7 | 1 | 9937 | 9.6% |
| | realTraffic | 7 | 1 | 2237 | 10.0% |
| | realTweets | 10 | 1 | 15863 | 9.9% |
| | artificialWithAnomaly | 6 | 1 | 4032 | 10.0% |
| NASA | SMAP | 54 | 25 | 2555/8070 | 12.83 % |
| | MSL | 27 | 55 | 1785/2730 | 10.53 % |
| SMD | machine-1 | 8 | 38 | 24296/24296 | 6.04 % |
| | machine-2 | 9 | 38 | 23691/24813 | 4.34 % |
| | machine-3 | 11 | 38 | 26424/26429 | 2.75 % |

E Contributions

C.B. proposed using ssGPs for time series AD and implemented ssGP and r-ssGP. Together with L.C. and F.X.A., he developed the robustification approach. He ran all benchmarking experiments and together with F.X.A. created all figures. He was one of the primary writers of the manuscript. F.X.A. proposed and derived ssGPFA. He implemented ssGPFA together with C.B. He proposed to constrain the model to have independent latent processes a posteriori. Together with J.G., he proposed the optimal closed form update of \mathbf{C} . He created the method allowing to gain explainability of the detected anomalies and implemented it. He was one of the primary writers of the manuscript.