Proceedings of Machine Learning Research 71:102–113, 2017 KDD 2017: Workshop on Anomaly Detection in Finance

# Binned Kernels for Anomaly Detection in Multi-timescale Data using Gaussian Processes

**Matthew van Adelsberg** MATTHEW.VANADELSBERG@CAPITALONE.COM
**Christian Schwantes** CHRISTIAN.SCHWANTES@CAPITALONE.COM
*1680 Capital One Dr McLean VA 22102*

## Abstract

Financial services and technology companies invest significantly in monitoring their complex technology infrastructures to allow for quick responses to technology failures. Because of the volume and velocity of signals monitored (e.g., customer transaction volume, API calls, server CPU utilization, etc.), they require sophisticated models of normal system behavior to determine when a component falls into an anomalous state. Gaussian processes (GPs) are flexible, Bayesian nonparametric models that have successfully been used for time series forecasting, interpolation, and anomaly detection in complex data sets. Despite the growing use of GPs for time series analysis in the literature, these methods scale poorly with the size of the data. In particular, data sets containing multiple timescales can pose a problem for GPs, as they can require a large number of points for training.

We describe a novel method for including long and short timescale information without including an impractical number of data points through the use of a binned process, defined as the definite integral over a latent Gaussian process. This results in a binned covariance function for the time series, which we use to fit and forecast data at multiple resolutions. The resulting models achieve higher accuracy with fewer data points than their non-binned counterparts, and are more robust to long tailed noise, heteroskedasticity, and data artifacts.

**Keywords:** Gaussian process, kernel, time series forecasting, anomaly detection

## 1. Introduction

Financial services and technology companies leverage complex technology infrastructures to deliver valuable digital experiences to customers. These corporations must monitor technology operations to identify interruptions in service and hardware/software failures, enabling rapid root cause analysis and mitigation before adverse customer impacts occur. In practice, operations monitoring involves contextual anomaly detection in time series data from a variety of sources, including customer transaction volume, software API calls, server CPU utilization, etc. Anomalies can be identified in a number of ways, for example, by fitting a model to training data and assessing misfit between the model forecast and future data, or by fitting sequential changepoint or fault detection models (e.g., Garnett et al., 2010).

A significant challenge in controlling the false positive rate of incident alerts is the presence of multiple timescales in the data, which appear anomalous unless an impractical amount of data are used for model training. Obvious drops or spikes in traffic volume on the minute or hour timescale are readily identified using a variety of methods (see, e.g., Woodall

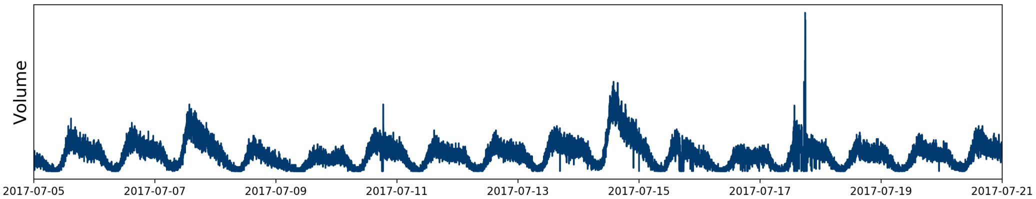

Figure 1: Financial transaction data exhibiting multiple timescales with hourly, daily, and weekly variation.

and Montgomery, 1999). However, successfully identifying more subtle faults might involve, for example, comparing minute level forecasts for several hours, or hour level forecasts over the course of a day, to new data and creating an alert when the model unsuccessfully captures the covariance structure of the underlying signal.

As an example, consider the time series of financial transaction data in Figure 1. These data exhibit noise on the minute timescale, with quasi-periodic fluctuations on the hour, day, and weekly timescales. Consider the increase in overall traffic on July 15, 2017; it is clear from examining the time series that the enhanced volume is not associated with an anomaly, but rather an overall weekly pattern. Imagine now that, on the morning of July 15, we create a forecast of the signal over the next few hours, conditioned only on hourly data from the past few days. In this case, there is no failure, but we would create a false alert, based on the increase in transaction volume over our prediction.

These kinds of false positives are inevitable when monitoring large enterprise technology infrastructures, in which we need to identify anomalies that occur in between multiple timescales of interest. As an example, a failure in one data center might lead to increasing transaction volume routed to a different data center. Such a failure might manifest as an increase in traffic over several hours before the backup data center fails under the enhanced transaction load; the underlying fault in the original data center could only be detected by identifying the deviation in time series data from the forecasted daily pattern.

For this reason, and due to the volume, velocity, and variety of actionable data in large enterprises, we require sophisticated models of system behavior to avoid generating an impractical number of alerts while identifying critical software and hardware failures. Gaussian processes (GPs) represent an extremely flexible, powerful modeling approach for capturing the covariance structure of data in a wide range of applications. A number of recent works discuss time series applications of GPs for anomaly detection and changepoint modeling (see, e.g., Adams and MacKay, 2007; Saatci, 2011; Roberts et al., 2012, and the references therein). Gaussian processes have also been used to create Bayesian nonparametric generalizations of standard state-space models for linear dynamical systems, with applications to, for example, stochastic volatility (e.g., Wu et al., 2014; Frigola-Alcalde, 2015). In addition, a number of authors have used GPs to model the correlations between time series, with applications to financial data (e.g., Wilson and Ghahramani, 2011; Wilson et al., 2012). A comprehensive introduction to the Gaussian process literature can be found in Rasmussen and Williams (2006)

Despite the promise of GPs for time series modeling, a major practical drawback is the unfavorable scaling of GP learning algorithms with the size of the data set. For $T$ data points, inference in GP models requires matrix decompositions that scale as $\mathcal{O}\left(T^3\right)$. For models in which time is the only variable, and is laid out on a grid with equal spacing, the complexity can be reduced to $\mathcal{O}\left(T^2\right)$ using Toeplitz methods (Zhang et al., 2005). More generally, other authors have introduced a variety of approximate methods for learning GPs which effectively downsample the data to reduce the effective scaling (e.g., Smola and Bartlett, 2001; Quiñonero Candela and Rasmussen, 2005; Snelson and Ghahramani, 2006; Wilson and Nickisch, 2015).

In applications of GPs to data with multiple variation time scales, the data burden can become restrictive. For example, the ability to forecast multiple hours ahead, taking into account daily and weekly variation, requires training data at the hourly scale over several weeks. This can quickly become restrictive, even when using Toeplitz methods or downsampling of the data. We present a simple alternative method for learning time series behavior on multiple, disparate timescales, leading to improved forecasts and concomitantly lower false positive alert rates, with significantly reduced data requirements.

Time series data for monitoring are often binned over specific timescales to increase signal-to-noise and facilitate understanding of patterns in the data. For example, aggregating CPU utilization over an hour time interval might yield a clear periodic daily pattern using many fewer data points than at a minute-level bin size. To exploit this intuition, we model the data with a latent, unaggregated time series, which is given a GP prior. We then define transformations of the data, consisting of aggregations over multiple, distinct timescales (e.g., minute, hour, day, etc.). The GP prior on the latent time series induces a GP on each of these aggregations. The kernel functions describing the covariance within and between the transformed quantities can be written as integrals over the bin size on the latent GP kernel. These integrals can be carried out by taking the Fourier transform of the latent kernel and applying the aggregation in the spectral domain. This procedure results in covariance and cross-covariance functions between data aggregated with different bin sizes. From the properties of the GP, posterior predictions for future data are calculated using linear combinations of the data at all timescales.

The outline of the rest of the paper is as follows: in §2, we describe the binned kernel approach, and derive practical methods for incorporating these into standard GP model training; in §3, we test our methodology on synthetic and actual transactional time series data and demonstrate the benefits of the binned kernel approach; finally, in §4, we discuss implications for anomaly detection and make connections between the binned approach and other models.

## 2. Methods

Our modeling approach assumes the existence of a latent function, $f$, of a single time variable $t$. We place a Gaussian process prior on $f$:

$$f|\mathbf{t}, \theta \sim \mathcal{N}\left(0, K\right), \tag{1}$$

where $\mathcal{N}$ is the multivariate normal distribution, $\mathbf{t}$ is a set of $T$ time values, $K_{ij} \equiv k(t_i, t_j|\theta)$, with $k$ denoting the kernel function, and $\theta$ the collection of hyperparameters for the GP kernel (c.f., Rasmussen and Williams, 2006). We assume that latent, unbinned data are generated from a normal sampling distribution:

$$y|f, \sigma^2 \sim \mathcal{N}\left(f, \sigma^2 I_T\right), \tag{2}$$

where $\sigma^2$ is the variance parameter for the Gaussian likelihood, and $I_T$ denotes the $T \times T$ identity matrix.

The standard approach for learning with Gaussian process models is to place a prior, $p(\theta)$, on the hyperparameters, and to marginalize over the function values:

$$p\left(\theta|\mathbf{t}, y\right) = \int df p(\theta, f|\mathbf{t}, y) = \mathcal{N}\left(y|0, K + \sigma^2 I_T\right) p(\theta), \tag{3}$$

where we obtain the result on the right due to the fact that the convolution of two Gaussians is a Gaussian. We can sample from the posterior distribution for hyperparameters in Equation (3) using a Markov Chain Monte Carlo algorithm (e.g., Hamiltonian Monte Carlo or slice sampling; see Neal 1998 and Neal 2000). The GP kernel function determines the class of functions with appreciable density in the prior, and is a primary place for the model builder to inject inductive bias into the specification (see, e.g., the discussion in Wilson, 2014). For example, one of the most common kernel functions is the squared-exponential kernel:

$$k(t_i, t_j|\eta^2, \ell^2) = \eta^2 e^{-(t_i - t_j)^2/2\ell^2}, \tag{4}$$

where the hyperparameters $\eta^2$ and $\ell^2$ determine the magnitude and scale of variation of the class of smooth functions with significant support in the prior.

For time series forecasting, an important kernel function is the locally periodic kernel:

$$k(t_i, t_j|\eta^2, w, \nu, \ell^2) = \eta^2 e^{-\sin^2[\pi\nu(t_i - t_j)]/w^2} e^{-(t_i - t_j)^2/2\ell^2}, \tag{5}$$

which is the combination of periodic and squared-exponential kernels. As shown below, this kernel has broad support for functions whose spectra contain strong contributions at frequencies that are multiples of $\nu$; these correlations vanish on the timescale $\ell$. The parameter $w$ governs the number of harmonics of $\nu$ that contribute appreciably to the Fourier transform of Equation (5); see §2.3 below. This covariance structure is critical in time series forecasting for signals that exhibit quasi-periodic behavior.

There are a wide variety of kernel functions that are useful for time series modeling, as well as recent literature on techniques for kernel learning (see, e.g., Rasmussen and Williams, 2006; Garnett et al., 2010; Wilson and Adams, 2013; Oliva et al., 2016). In the rest of this paper, we focus on the locally periodic kernel, and the challenges inherent in evaluating binned kernels based on it.

### 2.1. Binned Kernels

A linear transformation of a GP results in another GP (Rasmussen and Williams, 2006). If we define two binned functions with GP priors:

$$f_{\Delta}(t) \equiv \frac{1}{\Delta} \int_{t-\Delta/2}^{t+\Delta/2} y(\xi) d\xi, \tag{6}$$

$$f_{\Delta'}(t') \equiv \frac{1}{\Delta'} \int_{t'-\Delta'/2}^{t'+\Delta'/2} y(\xi) d\xi, \tag{7}$$

this results in GPs defined on intervals $t_{\Delta} \equiv (t - \Delta/2, t + \Delta/2)$, $t'_{\Delta'} \equiv (t' - \Delta'/2, t' + \Delta'/2)$. The cross-covariance between these functions is given by (e.g., Lawrence et al., 2006):

$$\bar{k}\left(t_{\Delta}, t'_{\Delta'}\right) = \frac{1}{\Delta\Delta'} \int_{t-\Delta/2}^{t+\Delta/2} \int_{t'-\Delta'/2}^{t'+\Delta'/2} k(\xi, \xi') d\xi d\xi', \tag{8}$$

where $k$ denotes the kernel function for the base GP, $f$, plus the contribution from the latent white noise kernel defined in Equation (2). This form for the cross-covariance between binned GPs has been used to build differentially private GP learning algorithms (Smith et al., 2016).

Given the noise model for the latent function in Equation (2), we can derive noise terms for the cross-covariance of the binned functions. To derive this relationship, consider a sequence of kernel functions of the form:

$$k_n(t, t') = \frac{n\sigma^2}{2} e^{-n|t-t'|}, \quad n = 1, 2, \ldots \tag{9}$$

Consider two intervals, $t_{\Delta}$ and $t'_{\Delta'}$. If we denote the amount of overlap between the two intervals as $\Delta_{\mathrm{o}}$, then substituting Equation (9) into (8) yields the following contribution from the latent noise:

$$\frac{1}{\Delta\Delta'} \int_{t-\Delta/2}^{t+\Delta/2} \int_{t'-\Delta'/2}^{t'+\Delta'/2} k_n(\xi, \xi')\, d\xi\, d\xi' = \frac{\Delta_{\mathrm{o}}\sigma^2}{\Delta\Delta'}. \tag{10}$$

If we take the limit as $n \to \infty$, we recover the standard latent noise kernel $\sigma^2\delta(t - t')$; since Equation (10) is independent of $n$, this is the contribution of the latent noise to the cross-covariance.

With the binned cross-covariance defined, we can train the kernel hyperparameters and derive the posterior predictive forecast using standard methods (e.g., Rasmussen and Williams, 2006). For example, to create an hourly forecast taking into account daily and weekly patterns, we might decompose our recent time series data into the 100 most recent hourly data points, along with the most recent 50 additional data points binned at half-day intervals. This allows us to learn hourly and weekly patterns using 150 data points, as opposed to taking, for example, 500 hourly data points over the previous three weeks. The binned data contain more information than if we simply took 150 randomly sampled time points over a three week period (see §3).

### 2.2. Spectral Evaluation of Binned Kernels

In general, it is difficult to evaluate the integral in Equation (8). Except in the case of the squared-exponential kernel (c.f., Smith et al., 2016), the integrals do not reduce to simple analytic or semi-analytic forms. Evaluating the double integral using standard numerical quadrature techniques is computationally too slow for MCMC algorithms.

It turns out, however, that the double integral can be evaluated efficiently if we have a simple form for the Fourier transform of the latent kernel. With the binned cross-covariance defined as in Equation (8), it can be shown that the following identity holds (see Appendix A):

$$\bar{k}(t_\Delta, t'_{\Delta'}) = \frac{1}{\Delta\Delta'} \int\limits_{-\infty}^{\infty} \mathrm{sinc}(\Delta f)\mathrm{sinc}(\Delta' f)\hat{k}(f) e^{i2\pi f(t-t')} df, \tag{11}$$

where $\hat{k}(f)$ is the Fourier transform of the latent kernel $k(t, t')$. From Bochner's theorem, we know that $\hat{k}(f)$ is proportional to a probability measure (e.g., Rudin, 1990). Thus, we can evaluate the integral in Equation (11) using Monte Carlo integration as:

$$\bar{k}(t_\Delta, t'_{\Delta'}) \approx \frac{\hat{k}(0)}{S\Delta\Delta'} \sum_{s=1}^{S} \mathrm{sinc}(\Delta f_s)\mathrm{sinc}(\Delta' f_s) \cos(2\pi f_s |t_\Delta - t'_{\Delta'}|), \tag{12}$$

where

$$f_s \sim P(f) = \hat{k}(f)/\hat{k}(0), \quad s = 1, \ldots, S. \tag{13}$$

This procedure is equivalent to using the method of random Fourier features for the approximation of kernels, taking into account the effect of the binning procedure (see Rahimi and Recht, 2009). This leads to efficient evaluation of the kernel and subsequent linear algebra for sampling the posterior distribution for the hyperparameters in Equation (3). If we expand the cosine term in Equation (12), the kernel evaluation can be written as:

$$\bar{k}(t_\Delta, t'_{\Delta'}) \approx \hat{k}(0)\phi(t_\Delta)^{\mathrm{T}}\phi(t'_{\Delta'}), \tag{14}$$

where

$$\begin{aligned}\varphi(t_\Delta)^{\mathrm{T}} \equiv [&\mathrm{sinc}(\Delta f_1)\cos(2\pi f_1 t_\Delta), \ldots, \mathrm{sinc}(\Delta f_S)\cos(2\pi f_S t_\Delta), \ldots, \\ &\mathrm{sinc}(\Delta f_1)\sin(2\pi f_1 t_\Delta), \ldots, \mathrm{sinc}(\Delta f_S)\sin(2\pi f_S t_\Delta)]/\left(\sqrt{S}\Delta\right).\end{aligned} \tag{15}$$

In this case, the covariance and cross-covariance matrices take the form:

$$K \approx \hat{k}(0)UU'^{\mathrm{T}}, \tag{16}$$

where

$$U'^{\mathrm{T}} \equiv \left[\varphi(t'_{\Delta',1}), \ldots, \varphi(t'_{\Delta',T'})\right], \tag{17}$$

and $T'$ is the number of points in the binned grid. This decomposition allows the linear algebra associated with sampling from Equation (3) to be performed in $\mathcal{O}\left(S^3 + S^2(T + T')\right)$ operations, since the calculations can all be performed in the primal space of $\varphi(t_\Delta)$, rather than constructing large Gram matrices (see the discussion in Oliva et al., 2016).

Thus, for any kernel function with a simple Fourier transform, we can efficiently evaluate the binned kernel function and sample the posterior distribution for the GP hyperparameters with linear algebra that scales linearly with the number of points in the time grid.

### 2.3. Binned Locally Periodic Kernel

Unfortunately, the locally periodic kernel does not have a simple form for its Fourier transform. As shown in Appendix B, the periodic part of the kernel in Equation (5) can be expanded in a Fourier series:

$$e^{-\sin^2[\pi\nu(t_i-t_j)]/w^2} = a_0/2 + \sum_{n=1}^{\infty} a_n \cos(2\pi n\nu|t-t'|), \tag{18}$$

with

$$a_n = 2e^{-1/2w^2} I_n(1/2w^2), \tag{19}$$

where $I_n$ is the modified Bessel function of the first kind. The Fourier series expansion can be used to show that the Fourier transform for the locally periodic kernel is:

$$\hat{k}(f) = (a_0/2)N\left(f|0,\sigma_\ell^2\right) + \sum_{n=1}^{\infty}(a_n/2)\left[N(f|n\nu,\sigma_\ell^2) + N(f|-n\nu,\sigma_\ell^2)\right], \tag{20}$$

where $\sigma_\ell \equiv 1/4\pi^2\ell^2$. We can then apply the spectral binning procedure to Equation (20), where the total number of frequency points required is $N \times S$, and $N$ is the component at which the Fourier series is truncated. For models with a wide range of values for $w$ and $\ell$, we have $NS \lesssim T$, where $T$ is the total number of binned data points. Thus, the binned kernel function and associated linear algebra for learning the kernel hyperparameters can be performed efficiently, despite the absence of an analytic form for the kernel Fourier transform.

## 3. Results

To test our approach, we generated synthetic data using a locally periodic kernel for the latent function, as described in §2, with hyperparameters $\eta^2 = 1, w = 2^{-1/2}, \nu = 1, \ell = 10$, over a time grid ranging from $(-3, 3)$ in (arbitrary) units of "days." In Figure 2, we show the resulting training (thin grey curve) and test (thick red line) data sets for three levels of the latent noise $\sigma^2$: $3\times10^{-4}$ (left panels), $10^{-2}$ (middle panels), and $3\times10^{-2}$ (right panels). We then fit this data using two approaches: 1) we sampled 36 intervals of 5 minutes from the training set at random and to build the small-bin model and 2) we replaced 18 of those 5 minute intervals with 4 hour intervals to build the large-bin model. Each model included the same number of data elements, but we hypothesized that the larger bins would allow for a more reliable forecast past a few minutes.

In the low noise scenario depicted in the left panels, the only difference between the small- and large-bin approaches is a slight reduction in uncertainty for the large-bin model (bottom left panel). However, as the latent noise increases from left to right, we see a significant improvement of the large-bin model over its small-bin counterpart.

We then performed a more systematic investigation of small bin vs large bin performance across a wide range of synthetic data sets. Keeping the $\eta^2, w, \nu$ hyperparameters fixed as above, we varied the signal-to-noise ratio $\eta/\sigma$ from 1 to $10^4$, taking 75 draws of synthetic data for each ratio. We then analyzed the data with binned and non-binned models as described above, and compared relative accuracy on the test data using the Mean Absolute

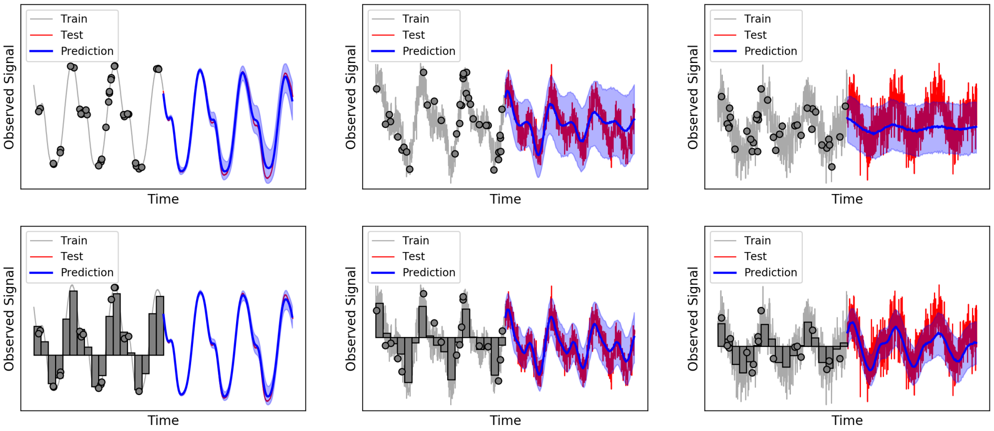

Figure 2: Test of binned kernel methodology on synthetic data. From left to right, panels show data generated with increasing values of the noise variance. For noisy data, the binned model significantly outperforms its non-binned counterpart using the same number of data points.

Scaled Error (MASE) metric (e.g., Frances, 2016). The results are shown in Figure 3, in which the performance of the small and large bin models are shown by the light green and dark red regions, respectively. It is clear from the figure that, for a wide range of function draws and signal-to-noise ratios, the large-bin model outperforms its small-bin counterpart using the same number of data points. For extremely low or high signal-to-noise ratios, there are smaller differences in performance between the two models.

After testing our methodology on synthetic data, we fit binned and non-binned models to data collected from financial transaction volume monitoring in a real world use case. The top panel of Figure 4 shows the data, split into a training set (solid black curve) and test set (solid red curve). The middle and bottom panels show the fit of the binned and non-binned models, respectively. For the non-binned approach, we randomly sampled 50 minutes in the training set and used the the locally periodic kernel (without binning) to create a ten day forecast. For the binned approach, we sampled 30 minute intervals as well as 20 four-hour intervals throughout the training set and used a binned locally periodic kernel to create the forecast. (Note: in both cases the kernel was defined to be the sum of two locally periodic kernels to account for the weekly and daily periodicities.)

Though the data are noisy, the binned model does a better job of capturing the underlying signal than the non-binned version for the same number of data points. This demonstrates one of the benefits of using binned data: aggregating data over larger bins tends to smooth out noise, making binned models more robust to time series exhibiting deviations from Gaussian noise (i.e., long tailed likelihoods), heteroskedasticity, and poor data quality.

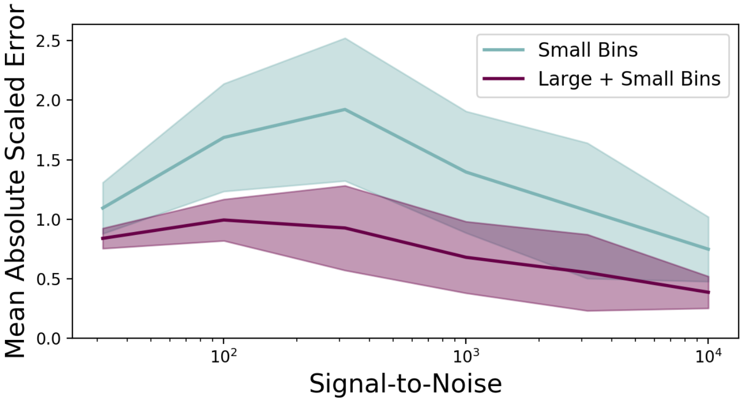

Figure 3: Systematic investigation of large- vs small-bin performance over a wide range of synthetic data generated by taking 75 function draws for each value of the signal-to-noise ratio determined by the latent function hyperparameters. Except at the extreme ends of low and high noise, the large-bin models outperform their small-bin counterparts on the Mean Absolute Scaled Error metric.

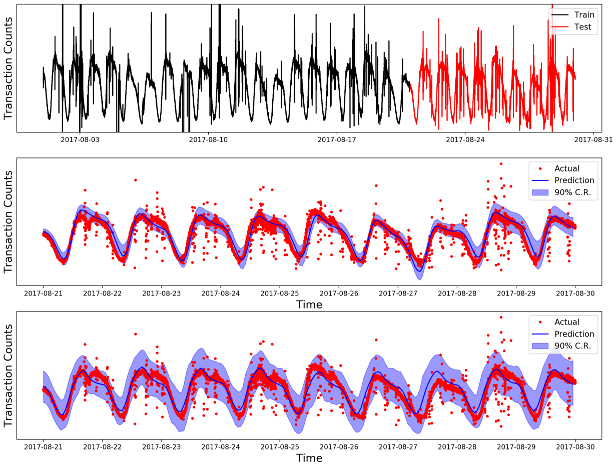

Figure 4: Test of model methodology on a real world transaction volume time series. The raw data are shown in the top panel, while the middle and bottom panels show results for the binned and non-binned models, respectively. The binned models do a much better job of capturing the underlying signal, despite significant, long-tailed noise in the raw signal.

## 4. Discussion

We have derived a novel approach to constructing binned kernels that capture multiple timescales in a unified modeling framework. Except at the extremes of low or high noise, models constructed using binned kernels have lower data requirements than their non-binned counterparts, improving the efficiency of fitting a GP as well as generating forecasts for multi-timescale data. In addition, due to the reduction in error for larger intervals, binned models can exhibit increased robustness with respect to long-tailed or heteroskedastic noise, as well as data artifacts.

Monitoring enterprise IT infrastructure can involve modeling hundreds to thousands of time series. Unless our models achieve extremely low false detection rates, more alerts will be generated than can be efficiently resolved. Thus, models with high forecast accuracy are required to simultaneously achieve reasonable false positive and negative rates, which is why we have focused on forecast accuracy here. The binned kernel approach will have the biggest impact on accuracy for subtle anomalies that occur gradually at the boundaries of multiple timescales (versus, e.g., drastic changes in output scale; see §4.4 of Garnett et al. 2010). Though this approach might only result in a small reduction in the absolute false positive rate, this could be crucial for reducing the absolute number of alerts to a manageable number. One additional consideration not included in this work is how to choose the size and location of each of the bins. We plan to systematically explore the effects of binned kernel techniques – including bin selection – on anomaly detection performance in a future work.

As discussed in §2, integrals over latent functions have been used to create differentially private GPs (Smith et al., 2016). In their paper, Smith et al. consider integrated kernels based on a squared-exponential kernel for a latent Gaussian process. Their calculations exploited the analytic properties of integrals over a Gaussian kernel. The work in our paper could be used to extend their results to create efficient differentially private models using a broader class of latent kernel functions.

Finally, we briefly mention a connection between the work in this paper and the spectral mixture kernel introduced by Wilson and Adams (2013). The spectral mixture kernel is derived by starting with a scale-location mixture of Gaussians in the Fourier domain, and taking the inverse Fourier transform to derive an analytic form for the kernel in the time domain. Because a scale-location mixture of Gaussians can approximate any distribution to arbitrary accuracy, a spectral mixture kernel with enough components can, in principle, accurately model a GP with any stationary kernel function.

The Fourier series representation of the locally periodic kernel is equivalent to an infinite spectral mixture kernel expansion. The form of the series shows that periodic behavior implies a restrictive set of constraints on the amplitudes and frequencies of the components of the spectral mixture. This suggests that it might be difficult in practice to train spectral mixture models on data with recurring patterns, due to the fact that kernels with periodic components occupy a very small region in the space of functions implied by all possible stationary kernels. Some initial experiments on the data sets presented in this paper suggest that this is indeed the case. This suggests a potential avenue for further research, in which hierarchical prior distributions on the amplitudes and frequencies of the spectral mixture kernel place additional probability mass on regions of kernel space that exhibit periodic behavior; such a process over kernels could make it easier to perform kernel learning on data sets containing periodic signals.

## References

R. Adams and D. MacKay. Bayesian online changepoint detection. Technical report, University of Cambridge, 2007.

P. Frances. A note on the mean absolute scaled error. *International Journal of Forecasting*, 32(1), 2016.

R. Frigola-Alcalde. *Bayesian Time Series Learning with Gaussian Processes*. PhD thesis, University of Cambridge, 2015.

R. Garnett, M. Osborne, S. Reece, A. Rogers, and S. Roberts. Sequential bayesian prediction in the presence of changepoints and faults. *The Computer Journal*, 53(9):1430–1446, 2010.

N. Lawrence, G. Sanguinetti, and M. Rattray. Modelling transcriptional regulation using gaussian processes. *Advances in Neural Information Processing Systems*, 2006.

R. Neal. Regression and classification using gaussian process priors. In J. Bernardo, editor, *Bayesian Statistics*. Oxford University Press, 1998.

R. Neal. Slice sampling. Technical report, University of Toronto, 2000.

J. Oliva, A. Dubey, A. Wilson, B. Póczos, J. Schneider, and E. Xing. Bayesian nonparametric kernel-learning. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, 2016.

J. Quiñonero Candela and C. Rasmussen. A unifying view of sparse approximate gaussian process regression. *Journal of Machine Learning Research*, 6, 2005.

A. Rahimi and B. Recht. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. *Advances in Neural Information Processing Systems*, 2009.

C. Rasmussen and C. Williams. *Gaussian processes for Machine Learning*. MIT Press, 2006.

S. Roberts, M. Osborne, M. Ebden, S. Reece, N. Gibson, and S. Aigrain. Gaussian processes for time-series modelling. *Philosophical Transactions of the Royal Society A*, 371(1984), 2012.

W. Rudin. *Fourier analysis on groups*. John Wiley and Sons, 1990.

Y. Saatci. *Scalable Inference for Structured Gaussian Process Models*. PhD thesis, University of Cambridge, 2011.

M. Smith, M. Zwiessele, and N. Lawrence. Differentially private gaussian processes. In *29th Conference on Neural Information Processing Systems*, 2016.

A. Smola and P. Bartlett. Sparse greedy gaussian process regression. *Advances in Neural Information Processing Systems*, 13, 2001.

E. Snelson and Z. Ghahramani. Sparse gaussian processes using pseudo-inputs. *Advances in Neural Information Processing Systems*, 18, 2006.

A. Wilson. *Covariance Kernels for Fast Automatic Pattern Discovery and Extrapolation with Gaussian Processes*. PhD thesis, University of Cambridge, 2014.

A. Wilson and R. Adams. Gaussian process kernels for pattern discovery and extrapolation. *ICML*, 2013.

A. Wilson and Z. Ghahramani. Generalised wishart processes. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 2011.

A. Wilson and H. Nickisch. Kernel interpolation for scalable structured gaussian processes (kiss-gp). *International Conference on Machine Learning*, 2015.

A. Wilson, D. Knowles, and Z. Ghahramani. Gaussian process regression networks. In J. Langford and J. Pineau, editors, *Proceedings of the 29th International Conference on Machine Learning*, 2012.

W. Woodall and D. Montgomery. Research issues and ideas in statistical process control. *Journal of Quality Technology*, 32, 1999.

Y. Wu, J. Hernandez-Lobato, and Z. Ghahramani. Gaussian process volatility model. *Advances in Neural Information Processing Systemss in Neural Information Processing Systems*, 27, 2014.

Y. Zhang, W. Leithead, and D. Leith. Time-series gaussian process regression based on toeplitz computation. *Optimization*, 21(126), 2005.