The New England Journal of Statistics in Data Science Volume 1, 84–94 (2023)
DOI: https://doi.org/10.51387/23-NEJSDS20

Detection of Anomalies in Traffic Flows with Large Amounts of Missing Data

Qing HE[1], Charles W. HARRISON[1], and Hsin-Hsiung HUANG*

Abstract

Anomaly detection plays an important role in traffic operations and control. Missingness in spatial-temporal datasets prohibits anomaly detection algorithms from learning characteristic rules and patterns due to the lack of large amounts of data. This paper proposes an anomaly detection scheme for the 2021 Algorithms for Threat Detection (ATD) challenge based on Gaussian process models that generate features used in a logistic regression model which leads to high prediction accuracy for sparse traffic flow data with a large proportion of missingness. The dataset is provided by the National Science Foundation (NSF) in conjunction with the National Geospatial-Intelligence Agency (NGA), and it consists of thousands of labeled traffic flow records for 400 sensors from 2011 to 2020. Each sensor is purposely downsampled by NSF and NGA in order to simulate missing completely at random, and the missing rates are 99%, 98%, 95%, and 90%. Hence, it is challenging to detect anomalies from the sparse traffic flow data. The proposed scheme makes use of traffic patterns at different times of day and on different days of week to recover the complete data. The proposed anomaly detection scheme is computationally efficient by allowing parallel computation on different sensors. The proposed method is one of the two top performing algorithms in the 2021 ATD challenge.

Keywords and phrases: Anomaly detection, Gaussian process, Spatiotemporal, High dimensional, Missing completely at random.

1. INTRODUCTION

Spatial-temporal datasets collected over time in different locations are often encountered in practice, and one emerging application of interest in spatiotemporal datasets is anomaly detection. For traffic flow data, anomaly detection is an important component in traffic operations and control since abnormal traffic events can greatly reduce traffic efficiency [30]. There is an increasing demand for automated, efficient, and universal anomaly detection methods as more traffic cameras are deployed to record road data [2]. Recent literature has developed machine learning and deep learning-based methods for detecting spatial-temporal anomalies. For example, in supervised deep anomaly detection, both normal and anomalous data are used to train a binary or multi-class classifier [2]. Kut and Birant [13] proposed a point anomaly detection algorithm using a clustering-based and density-based approach to discover clusters according to non-spatial, spatial, and temporal values of the objects. Although anomaly detection has been well studied in a variety of research and application domains, few are focused on sparse data with missingness which is a practical challenge encountered in a spatial-temporal setting and concerns external factors interfering with the data collection process [29, 25, 16]. The sparsity manifests itself in a number of ways including the introduction of additional measurement errors in the collected data observations or preventing the collection of the entire data observations. Missingness may not present many challenges if most of the data are observable, but when the majority of the data are missing, these algorithms may not be feasible.

Missing data are a common occurrence in many areas of research. Missingness in a dataset can be categorized as missing completely at random, missing at random, or missing not at random [14]. Data are missing completely at random if the failure to observe a value does not depend on any values of the response, either observed or missing, or any other observed values. As the observed data are just a random sample of the complete data and the outcomes do not affect the model for the missing data, the missing and observed data are not systematically different [5]. For missing at random, there might be systematic differences between the missing and observed data, but these can be entirely explained by other observed variables. Missing not at random occurs if missingness depends not only on the observed data but also on the unobserved (missing) values.

The focus of this paper is to propose a spatial-temporal anomaly detection framework to identify an anomalous traffic flow (i.e., the number of vehicles at a specific sensor location at a given time and weekday) when most of the data observations are missing due to external factors. Since the provided data are extremely sparse, with missing rates

*Corresponding author.
[1]Advised by the corresponding author.

ranging from 90% to 99%, using the data directly does not yield satisfying results. It is challenging to detect anomalous events because they rarely happen for most sensors, and even fewer are observed. The extreme scarcity of the incomplete dataset makes it more difficult to detect the occurrence of anomalous events directly. Our work contains the following contributions:

- A three-layer data engineering architecture is proposed for data transformation and augmentation based on spatial-temporal patterns.
- Gaussian process regression (GPR) is used to reconstruct the complete data and estimate features such as mean, standard deviation and percentiles.
- A logistic regression model is built to predict anomaly status where the cutoff is optimized through cross-validation.

The remainder of this paper is arranged as follows. In Section 2, we provide a summary of the challenges. In Section 3, we give a general introduction to GPR models. The proposed anomaly detection framework is introduced in 4. Evaluation of the proposed method and some key patterns identified from the study are shown in Section 5. The final section 6 provides a discussion and offers directions for future research.

2. OVERVIEW

2.1 Traffic Flow Dataset

This work is motivated by a challenging traffic data problem provided by the 2021 Algorithms for Threat Detection (ATD) program sponsored by the National Science Foundation (NSF) and the National Geospatial-Intelligence Agency (NGA) [1]. The complete dataset (i.e., without missing values) consists of hourly traffic flow (i.e., the number of vehicles) for 400 sensors (locations) in California from 2011 to 2020. A sample of the complete data is provided in Table 1.

The goal of the 2021 ATD challenge is to detect traffic flow anomalies at a sensor s at hour h on weekday w for an incomplete dataset where the majority of the data is not observed. A sample of the incomplete data is provided in Table 2. In fact, the ATD organizers downsampled each sensor by a particular sampling rate (i.e., one minus the missing rate): 1%, 2%, 5%, or 10%. The incomplete dataset has the same form as the complete dataset except that a large number of rows are missing as depicted in Table 2.

The definition of an anomaly is determined by the ATD program organizers. All analyses are at the granular level of sensor s, hour h, and weekday w, which is called a slice (see definition 2.1). The traffic flow data are detrended for each slice as the anomaly definition assumes stationarity. Specifically, a linear trend model is fit on the complete hourly data over time, and its predicted linear trend is then subtracted from the original data. The average traffic flow is added back to the resulting data. We refer to the dataset that results

Table 1. A sample of the complete dataset of hourly traffic flow. This dataset is detrended within each slice (see Definition 2.1) and used to define the true labels, but only a random subsample of this dataset is provided for analysis.

Sensor	Date	Weekday	Hour	Traffic
1	2011/1/1	Saturday	0	2240
1	2011/1/1	Saturday	1	1835
1	2011/1/1	Saturday	2	1580
⋮	⋮	⋮	⋮	⋮
400	2020/12/31	Thursday	22	320
400	2020/12/31	Thursday	23	266

Table 2. A sample of the incomplete dataset that is provided to build an algorithm for anomaly detection with a sampling rate (the last column in the table) at 1%, 2%, 5%, or 10% for each sensor.

Sensor	Date	Weekday	Hour	Traffic	Rate
1	2011/1/1	Saturday	16	6278	1%
1	2011/1/19	Wednesday	17	6302	1%
⋮	⋮	⋮	⋮	⋮	⋮
400	2020/12/30	Wednesday	21	388	5%
400	2020/12/30	Thursday	19	466	5%

from the aforementioned steps as the complete detrended dataset D.

Let the detrended traffic flow be denoted by f. We refer to the complete detrended data (i.e., no missing values) concerning sensor s at hour h on weekday w as the "slice" which has the following definition.

Definition 2.1 (slice)**.** Let i be the ith row in the complete detrended dataset D. The slice refers to the complete detrended traffic flow data (i.e., no missing values) with respect to sensor s at hour h on weekday w:

$$\begin{aligned} D_{s,h,w} &:= \left\{ D_f^i : D_s^i = s \cap D_h^i = h \cap D_w^i = w \cap D_o^i = 1 \right\} \\ &:= \left\{ D_{s,h,w}^{i'} \right\}_{i'=1}^{i'=n_{s,h,w}} \end{aligned}$$

where D_s^i is the sensor index number in ith row, D_h^i is the hour value in the ith row, D_w^i is the weekday value in the ith row, $D_o^i = 1$ if the observation is sampled and $D_o^i = 0$ otherwise., and $n_{s,h,w} = |D_{s,h,w}|$.

Consequently, the true mean traffic flow for the slice is:

$$\mu_{s,h,w} = \frac{1}{n_{s,h,w}} \sum_i D_{s,h,w}^i \tag{2.1}$$

where $D_{s,h,w}^i$ is the ith traffic flow for sensor s at hour h on weekday w (i.e. multiple values collected from 2011 to

2020). Similarly, the true standard deviation for the slice is

$$\sigma_{s,h,w} = \sqrt{\frac{\sum_i (D^i_{s,h,w} - \mu_{s,h,w})^2}{n_{s,h,w}}} \tag{2.2}$$

The true anomaly label is determined by the ATD program organizers and is defined as the following:

$$I^i_{s,h,w} = \begin{cases} 1 \text{ (anomaly)}, & |D^i_{s,h,w} - \mu_{s,h,w}| \geq 3\sigma_{s,h,w} \\ 0 \text{ (normal)}, & |D^i_{s,h,w} - \mu_{s,h,w}| < 3\sigma_{s,h,w} \end{cases}.$$

In this project, the complete dataset D is not available. The sampling rate is provided for each sensor. As the definition of anomaly infers, the anomaly status is defined based on the complete dataset, but only the incomplete data are available to predict the anomaly. The purpose of this study is to use the incomplete data to accurately predict anomalies whose status is determined using the complete data.

3. METHODOLOGY

3.1 Gaussian Process Regression

Gaussian Process Regression (GPR) models are non-parametric kernel-based models based on the assumption that each observed response value y_i is sampled from a multivariate normal distribution. Different from linear regressions, GP regression models are built on random processes with a Gaussian prior instead of a parametric formulation for the latent function. It also provides uncertainty measures over predictions In GPR models, the relationship between a latent function $f(x_i)$ and the target (response) y_i can be written as follows:

$$y_i = f(x_i) + \epsilon_i,\ \epsilon_i \sim \mathrm{N}(0, \sigma^2),$$

where the random noise ϵ_i is normally distributed with mean 0 and variance σ^2. The covariance function of the response in a GPR is $\mathrm{cov}(y) = K + \sigma^2 I$ where the entry at the i-th row and j-th column of K, $K_{i,j} = k(x_i, x_j)$. The covariance matrix characterizes correlations between different responses in the process. If x_i and x_j have similar kernel function values, then their GPR outputs, $f(x_i)$ and $f(x_j)$, shall also be similar. For some new points X_*, the prediction $f(X_*)$ can be obtained from the joint distribution of f on the input X and f_* on the new points X_*, a simplified notation of $f(X_*)$ as follows:

$$\begin{bmatrix} f \\ f_* \end{bmatrix} \sim \mathrm{N}\left(\begin{bmatrix} m(X) \\ m(X_*) \end{bmatrix}, \begin{bmatrix} K & K_* \\ K_*^T & K_{**} \end{bmatrix}\right),$$

where $K = K(X,X)$, $K_* = K(X, X_*)$, $K_{**} = K(X_*, X_*)$ is a covariance matrix, whose entries are given by the covariance function (kernel), $K_{i,j} = k(x_i, x_j)$, and $m(X) = m(X_*) = 0$ for convenience. The posterior predictive distribution is obtained by marginalizing out the training set latent variables [21]:

$$p(f_*|y) = \int p(f, f_*|y) df = \frac{1}{p(y)} \int p(y|f) p(f, f_*) df,$$

and it can be written as

$$p(f_*|y) = N(K_{f_*,f}(K_{f,f} + \sigma^2 I)^{-1} y, K_{f_*,f_*} - K_{f_*,f}(K_{f,f} + \sigma^2 I)^{-1} K_{f,f_*}).$$

Notice that the covariance does not depend on the observed output y but only on the inputs X and X_*.

Recently, GPs have become a popular method for non-parametric modeling, and there are several reasons for using GPs. First, the locations of the observations used to train the model do not necessarily correspond to the locations of the points for which we wish to investigate the function at [22]. Inference for the GP at a set of locations X_* that differ from those corresponding to X can be achieved by evaluating the posterior distribution of $f(X_*)$. Second, there are various kernel functions that can be proposed to meet the modeling expectations. For example, if we believe the informativeness of past observations in explaining current data is a function of their time difference, the squared exponential function can be used in the covariance function, i.e., $f(x_i, x_j) = h^2 \exp[-(\frac{x_i - x_j}{\lambda})^2]$, where hyperparameters h and λ control the output scale of the function and the input, or time, scale. Kernels can also be combined together (e.g., addition and multiplication) to develop flexible non-linear correlations.

Other non-parametric models, such as splines are also commonly used for spatiotemporal data analysis as they rely on fewer assumptions about the underlying data generating mechanisms. Although spline models can implement complex nonlinear functions, they are less efficient in modeling the effects of covariate interactions when compared to GPs.

Approximate Bayesian computation (ABC) has been used widely for parameter inference in ecology and biology as it does not require the specification of a likelihood function [8, 4]. The ABC approach estimates the posterior distributions of parameters by simulation data with parameters sampled from the prior distribution. A value for the tolerance (ϵ) is pre-specified to determine if the distance between a simulation dataset and the observed dataset is less than or equal to ϵ, it is retained. Otherwise, it is discarded.

GPs have been used as a replacement for supervised neural networks in nonlinear regression and extended for classification [26, 20]. Previous research has shown that these two approaches were equivalent for a neural network model with a single-layer fully-connected neural network in the limit of infinite width [17, 27, 28]. For such a neural network with independent and identically distributed (i.i.d.) random parameters, as a result of an affine transformation of the final hidden layer, each scalar output contains a sum of i.i.d.

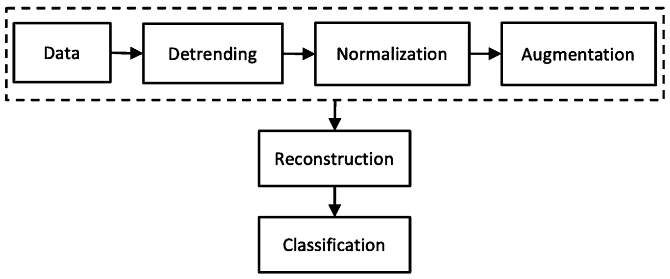


Figure 1: The proposed anomaly detection framework.

terms. According to the Central Limit Theorem, the neural network's computed function is drawn from a generalized projection (GP) when the width is infinite [27]. Multilayer Perceptron (MLP) neural networks are popular in capturing nonlinear relationships. It has been shown that a MLP neural network and GP have similar behavior when the number of hidden neurons tends to infinity and weight decay is employed. Based on such correspondence, it is possible to provide exact Bayesian inference for neural networks with infinite width by evaluating the corresponding generalized partials. There are kernel functions designed to mimic multilayer random neural networks, but they are only available outside of a Bayesian network and have not yet been identified as covariance functions for GP in previous work [27].

4. ANOMALY DETECTION FRAMEWORK

The proposed anomaly detection framework consists of the following steps which are discussed in detail in the subsections below. First, we pre-process the observed data by detrending and normalizing the data of each slice, and then we augment the incomplete data by transforming the observed traffic flow from temporal neighbors (e.g., prior and post hours as described in Section 4.1). Then, GPR models are built to recover the complete data based on the augmented data as described in Section 4.2 and 4.3. The GPR model is fit at the level of the slice (sensor, hour, and weekday) by exploiting the temporal correlations. With the reconstructed complete traffic flow, the means and standard deviations of the fitted traffic flow are computed for all slices. Finally, the anomaly and non-anomaly classification algorithm is built using logistic regression models, as described in Section 4.4. These steps are illustrated in Figure 1, where the data engineering step is included in the dashed rectangle. As the downsampled data have four different sampling rates (i.e., 1%, 2%, 5%, and 10%), we apply the proposed model to each sampling rate. A detailed algorithm for the proposed anomaly detection framework is in Algorithm 1.

4.1 Data Engineering

In this subsection, we describe how to pre-process the incomplete data which includes detrending the data in the same way as the ATD organizer, normalizing the data, and augmenting the data from similar groups: a) nearby hours of the same day and b) same hour of other weekdays. We first detrend the traffic flow following the same strategy described in Section 2.1. The resulting detrended data is denoted by d, and $d_{s,h,w}$ is the detrended total flow for slice (s, h, w).

In the normalization step, instead of normalizing the data based on the slice of s, h, and w which has high sparsity, the day of week is aggregated into weekdays or weekends to increase the sample size for normalization. A weekend indicator w_c is created to recode the day of week variable to a dichotomous variable with value 1 indicating weekends

Algorithm 1: The Proposed Anomaly Detection Algorithm.

```
1  Model Training
   Input: Downsampled traffic flow data $d$
2  for $(s,h,w) \in S$ where $S$ is the set of slices $(s,h,w)$ in $d$ do
3      Detrend $d_{s,h,w}$ as described in Section 2.1
4  for $(s,h,w_c) \in S^*$ where $w_c$ is the weekend indicator and $S^*$ is the set of slices $(s,h,w_c)$ in $d$ do
5      Normalized the detrended data for $d_{s,h,w_c}$
6  for $i = 1,\dots,n$ where $n$ is the size of $d$ do
7      Let $s$, $h$, and $w$ be its corresponding sensor ID, hour, and weekday
8      for $j = -k, -k+1, \dots, k-1, k+1$ and $j \neq 0$ do
9          Compute $\bar{d}'_{s,h+j,w}$ and $s_{d'_{s,h+j,w}}$ for slice $(s,h+j,w)$
10         Transform the total flow from using Equation (4.2)
11     for $w'$ s.t. $w_c(w') = w_c(w)$ do
12         Compute $\bar{d}'_{s,h,w'}$ and $s_{d'_{s,h,w'}}$ for slice $(s,h,w')$
13         Transform the total flow using Equation (4.3)
14 for $r = 1\%, 2\%, 5\%, 10\%$ do
15     for $(s,h,w)$ s.t. sampling rate of sensor $s = r$ do
16         Predict the missing total flow on the augmented data $y'_{s,h,w}$ with GPR model
17         Compute the mean $\hat{\mu}_{s,h,w}$, $\hat{\sigma}_{s,h,w}$, $p_5$, $p_{25}$, $p_{50}$, $p_{75}$, and $p_{95}$ of the reconstructed complete data
18     Predict the anomaly probability leveraging $d'$, $p'_5$, $p'_{25}$, $p'_{50}$, $p'_{75}$, and $p'_{95}$ as described in Section 4.4
19     Tune the cutoff $c$ using grid search to optimize the F1 Score
20 Model Testing
   Input: Downsampled traffic flow data $d_*$
21 for each sampling rate in the test dataset do
22     Follow the same data engineering and GRP steps as the training dataset
23     Use the fitted logistic regression model from training to estimate the probability of being anomalous
24     Use tuned $c$ to determine the anomalous status
25     Compute the test F1 score using Equation (5.1)
```

and 0 indicating weekdays,

$$w_c = \begin{cases} 1 & D_w^i \in \{\text{Sat., Sun.}\} \cap D_o^i = 1, \\ 0 & D_w^i \in \{\text{Mon., Tue., Wed., Thu., Fri.}\} \cap D_o^i = 1. \end{cases}$$

Then we normalize the data for each slice of s (sensor), h (hour), and w_c (weekdays or weekends). That is, data at the same locations and hours of weekdays are pooled together, and data at the same locations and hours of weekends are pooled together for normalization as follows,

$$d'^{\,i}_{s,h,w_c} = \frac{d^i_{s,h,w_c} - \bar{d}_{s,h,w_c}}{s_{d_{s,h,w_c}}}$$

where $\bar{d}_{s,h,w_c}$ and $s_{d_{s,h,w_c}}$ are the mean and standard deviation of slice (s,h,w_c). This approach is advantageous as compared to normalizing the data based on the slice of s, h, and w since there are only a few points available for each slice. It also works better than normalizing based on the slice of s and h because the traffic flows of weekdays and weekends are significantly different and thus not suitable for pooling.

In the data augmentation step, it is assumed that the traffic flow data is smooth so that observed data can be duplicated to other slices. We augment data from similar groups:

- nearby hours of the same day.
- same hour of other weekdays.

First let $d'_{s,h,w}$ be the detrended and normalized data for slice (s,h,w) from the downsampled data d, $d'^i_{s,h,w}$ be the its i-th observation. Denote the augmented data of slice (s,h,w) by $y'_{s,h,w}$ and the initial values are the same as $d'_{s,h,w}$ (i.e., set $y'_{s,h,w} = d'_{s,h,w}$ before the augmentation starts). To augment the data from nearby hours of the same day, let k be the parameter that determines the size of nearby hours to predict. Then, the traffic flow data are transformed as follows for the previous hours $h-1,\ldots,h-k$ and post hours $h+1,\ldots,h+k$:

$$\begin{aligned} y'^{(n+1)}_{s,h+j,w} &= g_1(d'^i_{s,h,w}) \qquad (4.1)\\ &= \frac{d'^i_{s,h,w} - \bar{d}'_{s,h,w}}{s_{d'_{s,h,w}}} \times s_{d'_{s,h+j,w}} + \bar{d}'_{s,h+j,w}, \end{aligned}$$

where $j = \pm 1,\ldots,\pm k$, n is the current sample size of $y'_{s,h+j,w}$ and updated after each augmentation step, $\bar{d}'_{s,h,w}$ and $s_{d'_{s,h,w}}$ are the sample mean and standard deviation of the data for slice (s,h,w), and $\bar{d}'_{s,h+j,w}$ and $s_{d'_{s,h+j,w}}$ are the sample mean and standard deviation of the data for slice $(s,h+j,w)$. Notice that in this step, we do not consider the weekday/weekend variable. Besides, we use grid search to find the optimum value of k (e.g., neighboring day window size for augmentation) for each sampling rate. The downsampled dataset with a lower sampling rate reasonably needs a larger value of k to achieve better results.

Similarly, the observed data from the same hour on other weekdays are also transformed for data augmentation. For slice (s,h,w), its traffic flow data are also used for other days of the week that have the same weekday category (weekday vs. weekend), i.e., slice (s,h,w') s.t. $w_c(w') = w_c(w)$. The transformation formula is described as follows:

$$\begin{aligned} y'^{(n+1)}_{s,h,w'} &= g_2(d'^i_{s,h,w}) \qquad (4.2)\\ &= \frac{d'^i_{s,h,w} - \bar{d}'_{s,h,w}}{s_{d'_{s,h,w}}} \times s_{d'_{s,h,w'}} + \bar{d}'_{s,h,w'} \qquad (4.3) \end{aligned}$$

where n is the current sample size of $y'_{s,h,w'}$ and updated after each augmentation step, $\bar{d}'_{s,h,w}$ and $s_{d'_{s,h,w}}$ are the sample mean and standard deviation of the data for slice (s,h,w), $\bar{d}'_{s,h,w'}$ and $s_{d'_{s,h,w'}}$ are the sample mean and standard deviation of the data for slice (s,h,w').

When the augmentation process completes, $y'_{s,h,w}$ contains the detrended data from the original incomplete dataset, transformed data from the neighboring hours of the same day, and transformed data from other similar days of week at the same hour and same week.

$$y'_{s,h,w} = d'_{s,h,w} \cup g_1(d'_{s,h\pm j,w}) \cup g_2(d'_{s,h,w'})$$

where $j = \pm 1,\ldots,\pm k$ and $w_c(w') = w_c(w)$. The augmentation strategy can be extended to incorporate spatial patterns by using data from the closest stations on the same day and same hour. It is not used in our project as it does not improve the model's performance.

4.2 Gaussian Process Regression Models

Although the data augmentation step has increased the sample size to a certain degree, it is based on observed data and does not infer what the unobserved looks like. Temporal patterns could be further explored to get a full picture of the data. The purpose of GPR models is to recover the unobserved traffic flow from the augmented data $d'_{s,h,w}$. To reconstruct the complete traffic flow data, we apply the fitted GPR models for each slice to predict the unobserved traffic flow from the posterior predictive distribution. With the augmented data $y'_{s,h,w}$, we apply GPR to recover the complete traffic flow data from 2011 to 2020 based on its date information, denoted by $t_{s,h,w}$. The dependent variable in the GPR is the traffic flow, and the dependent variable is the time index x, a new feature introduced to recover the time series pattern of the data. To find x, we first list all the possible dates that correspond to the hour and weekday value in slice (h,w):

$$T_{h,w} = \{\text{dates between 2011 and 2020} : T_h = h \cup T_w = w\}.$$

Let $N_{h,w}$ be the size of $T_{h,w}$. Then the time index for i-th observation in slice (s,w,h) $x^i_{s,h,w}$ for $y'_{s,h,w}$ can be identified by matching its dates with T as follows,

$$x^i_{s,h,w} = \{j : T^j_{h,w} = t^i_{s,h,w}\},$$

where $t^i_{s,h,w}$ is the date of i-th observation in slice (s, h, w). We have a collection $X_{s,h,w} \subset \{1, \dots, N_{h,w}\}$ by definition. Let $n_{s,h,w}$ be the size of $X_{s,h,w}$. For the set of observed data $G = \{(x^i_{s,h,w}, y'^i_{s,h,w}), i = 1, \dots, n_{s,h,w}\}$, we use the GPR to model the relationship between the traffic flow and the time index on the augmented data. Assume that

$$y'_{s,h,w} = f(x_{s,h,w}) + \epsilon, \epsilon \sim \mathrm{N}(0, \sigma^2 I).$$

A GP prior is placed over $f(x_{s,h,w})$ and simplifying the notation by removing the subscript (s, h, w) as follows:

$$p(f(x)|x) = \mathrm{N}(0, K)$$

where K is the covariance matrix determined by the kernel function. In this project, the rational quadratic kernel is implemented,

$$k_{\alpha,l}(x_i, x_j) = \left(1 + \frac{d(x_i, x_j)^2}{2\alpha\ell^2}\right)^{-\alpha}, \tag{4.4}$$

where α is the scale mixture parameter which determines the weighting of large or small-scale variations, l is the length scale of the kernel, and $d(\cdot, \cdot)$ is the Euclidean distance. The details of the kernel selection are discussed in the following section.

There are several advantages of recovering the complete sequence. First, the true mean and standard deviation could be approximated from the recovered data and fed into the classification process. Second, with the reconstructed complete time series data, other features such as the 5-th, 25-th, 50-th, 75-th, and 95-th percentiles can be computed and used when building the final prediction model. These summary statistics might not be represented well by the original incomplete data which have less than five observations $(n < 5)$ on average for each slice.

4.3 Kernel Selection

The choice of kernel profoundly affects the performance of a GP on a given task [23] as it controls properties of latent functions including smoothness and periodicity. In practice, sophisticated kernels are often achieved by composing a few standard kernel functions, which may also lead to overfitting. Commonly used kernels include the Gaussian radial basis function (RBF) kernel, rational quadratic kernel, expsine kernel, and Matérn kernel.

- The RBF kernel is also known as the squared-exponential kernel and takes the form as follows:

$$k_l(x_i, x_j) = \exp\left(-\frac{d(x_i, x_j)^2}{2l^2}\right)$$

  where l determines the length scale of the associated hypothesis space of functions. A large value of γ shrinks the covariance between nearby sets of observations, which results in a more smooth output.
- The rational quadratic kernel is described in Equation 4.4.
- The expsine kernel is in the form

$$k_{l,p}(x_i, x_j) = \exp\left(-\frac{2\sin^2(\pi d(x_i, x_j)/p)}{l^2}\right)$$

  where l is the length scale of the kernel, p is the periodicity of the kernel.
- The Matérn kernel is a generation of the Gaussian RBF kernel and given by:

$$k_{\nu,l}(x_i, x_j) = \frac{1}{\Gamma(\nu)2^{\nu-1}}\left(\frac{\sqrt{2\nu}}{l}d(x_i, x_j)\right)^{\nu} K_\nu\left(\frac{\sqrt{2\nu}}{l}d(x_i, x_j)\right)$$

  where $K_\nu(\cdot)$ is a modified Bessel function and $\Gamma(\cdot)$ is the gamma function.

In this project, we propose a few candidate kernels (i.e., the RBF kernel, rational quadratic kernel, expsine squared kernel, and Matérn kernel) and then use the cross-validation method to compute the mean absolute error (MAE) of the traffic flow for each candidate kernel. Specifically, we first random sample 20 different sensors and used the 3-fold cross-validation with five repetitions through the Python Scikit-learn package [19] to obtain the training and test indices for each slice (s, h, w). The GP regression is fit on the training dataset, and predictions are performed for the test dataset. The prediction errors from all the slices of the 20 sensors are collected to compute the MAE for each candidate kernel. The MAE comparison is shown in Table 3. The RBF, rational quadratic, and Matérn have comparable results, while ExpSine squared kernel gives a much higher MAE. Among the four sampling rates, the rational quadratic has the lowest MAE compared to RBF and Matérn, and is therefore selected to be the kernel function in GPR.

We use the Python Scikit-learn package [18] to optimize the hyperparameters in the kernel function, which are tuned in order to maximize the log-marginal-likelihood. Specifically, we use 0.1 and 1 as initial values for α and l, respectively. And the bounds are $\alpha, l \in (0.1, 10)$.

Table 3. The cross-validation MAE of four different kernels in GPR.

Sampling	RBF	Rational Quadratic	ExpSine Squared	Matérn
1%	0.760	0.723	38.062	0.753
2%	0.697	0.620	36.849	0.678
5%	0.666	0.575	10.031	0.648
10%	0.598	0.490	1.257	0.567
overall	0.602	0.680	21.549	0.661

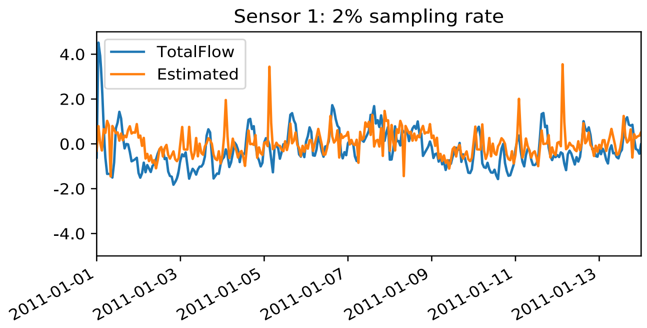


Figure 2: Actual traffic flow vs. the estimated flow using GP.

In Figure 2, we show the reconstructed time series for an arbitrary sensor (sensor 1) using GPR with a rational quadratic kernel. Sensor 1 has a 2% sampling rate. The blue curve represents the complete traffic flow and the orange is the reconstructed total flow. We only show the first two week's estimation (2011-01-01 to 2011-01-14) for better resolution. Overall, the estimated total flow resembles the true values, but it does not fully recover the true distribution due to the sparsity of the observed data.

4.4 Predictions via Logistic Regression

The incomplete traffic flow data has been detrended and normalized in the previous steps. There are additional features from the complete reconstructed data using GPR (i.e., estimated mean, standard deviation and percentiles). In this section, we will use d' (i.e., the detrended and normalized downsampled data), as well as features from GPR to build a logistic regression model for each sampling rate. All variables are normalized using the estimated mean and standard deviation of its corresponding slice. For example, data for slice (s, h, w), $d'^{\,i}_{s,h,w}$ is normalized as follows:

$$z^i_{s,h,w} = \left| \frac{d'^{\,i}_{s,h,w} - \hat{\mu}_{s,h,w}}{\hat{\sigma}_{s,h,w}} \right| \tag{4.5}$$

where $i = 1, \ldots, |d'_{s,h,w}|$, and $\hat{\mu}_{s,h,w}$ and $\hat{\sigma}_{s,h,w}$ are the mean and standard deviation of the reconstructed complete traffic flow for slice $(\{s, h, w)$ from GPR. Denote the normalized data of d' by z. Features from the reconstructed data are also normalized in the same fashion. For example, the 5-th percentile w.r.t. the reconstructed data for slice (s, h, w) is normalized as follows:

$$p'_{s,h,w;5} = \left| \frac{p_{s,h,w;5} - \hat{\mu}_{s,h,w}}{\hat{\sigma}_{s,h,w}} \right|$$

With the normalized data $\{z, p'_5, p'_{50}, p'_{75}, p'_{95}\}$, we train the logistic regression for each sampling rate. Define N_j be the size of sampling rate j where $j \in \{0.01, 0.02, 005, 0.1\}$. Omitting the slice notation $\{s, h, w\}$, the logistic regression model for sampling rate j can be written as

$$\log\left(\frac{p^i}{1-p^i}\right) = \beta^j_0 + \beta^j_1 z^i + \beta^j_2 p'^i_5 + \beta^j_3 p'^i_{25} \tag{4.6}$$
$$+ \beta^j_4 p'^i_{50} + \beta^j_5 p'^i_{75} + \beta^j_6 p'^i_{95} \tag{4.7}$$

where $i = 1, 2, \ldots, N_j$.

The logistic regression model returns the probability that a traffic flow is anomalous, and a cutoff threshold c_j needs to be specified to determine the anomaly status for each sampling rate j where $j \in \{0.01, 0.02, 005, 0.1\}$. A grid search is used to choose the optimal value of c_j. Note that the three standard deviation method to find anomalies is a special case for logistic regression when $\beta_0 = \beta_2 = \beta_3 = \beta_4 = \beta_5 = \beta_6 = 0$ and cutoff becomes $\frac{e^{3\beta_1}}{1+e^{3\beta_1}}$ as $z^i = 3$. So, the logistic regression model encompasses the three-standard deviation criterion. But it is more flexible than the three-standard deviation criterion because it allows the tuning process for c_j to obtain better prediction results.

Once the optimal values are selected, an observation is classified as an anomaly if its predicted probability of being anomalous is greater than c. For example, the optimal value of c was found to be 0.31 for a sampling rate of 0.1 so that an observation with predicted anomaly probability ≥ 0.31 is classified as an anomaly. This is defined in the formula below.

$$I(z^i_{s,h,w}) = \begin{cases} 1 \text{ (anomaly)}, & \hat{p}^i_{s,h,w} \geq c_j \\ 0 \text{ (not anomaly)}, & \hat{p}^i_{s,h,w} < c_j \end{cases} \tag{4.8}$$

where c_j is the optimal value for sampling rate j that sensor s corresponds to. Then we can compute the corresponding F1 score using Equation (5.1).

5. RESULTS AND DISCUSSION

5.1 Assessment of Predictive Performance

The classification results were evaluated with F1 score which is the metric adopted by the challenge organizer. It is defined as follows:

$$\text{F1} = \frac{\text{true positive}}{\text{true positive} + 0.5(\text{false positive} + \text{false negative})}. \tag{5.1}$$

F1 score is widely used in many applications such as image classification and documentation classification. Unlike classification accuracy which is the proportion of the number of correct predictions from all predictions made, F1 score conveys the balance between precision and recall. For example, in the imbalanced setting when negative cases dominate, the classification accuracy would be high by simply labeling all observations negative. However, the F1 score would be 0 as the predictions for the negative observations are incorrect.

To test the performance of the proposed method, we randomly sample 200 for training and the remaining 200 sensors

Table 4. F1 scores on the test dataset by sampling rates. XGBoost-1 is the comparison model that replaces GP with XGBoost for the proposed model to reconstruct the complete total flow, while XGBoost-2 is the comparison model that replaces logistic regression with XGBoost to predict the anomaly status. The Baseline model used the three-standard deviation rule based on the downsampled data. For the ABC approach, we estimate the parameters from the posterior distribution and then use a three standard deviation rule to find anomalies. The proposed model consistently outperforms the comparison models.

Test Set F1 Score					
Rate	Proposed	XGBoost-1	XGBoost-2	ABC	Baseline
1%	0.5505	0.4074	0.3838	0.2651	0.0000
2%	0.6588	0.6779	0.6153	0.4758	0.2010
5%	0.7382	0.4000	0.7018	0.4844	0.6078
10%	0.8170	0.8249	0.7913	0.5785	0.7570

for testing. The proposed algorithm is run in parallel at the sampling rate level. Within each sampling rate run, it can be further paralleled at the slice level for GPR hyperparameter tuning, which is comparably the most computationally expensive part among all components. Overall, the proposed algorithm is efficient and takes about 1 hour for training and 7 minutes for testing. The test F1 score for each sampling rate is given in Table 4. We also compute the F1 score on the test dataset using the baseline algorithm provided by NSF, which labels an observation as an anomaly if the detrended traffic flow is above or below three standard deviations from the observed mean.

Although this paper focuses on the use of the Gaussian Process in the anomaly detection framework (see Figure 1), other methods may be used to reconstruct the complete total flow. One method applied is stochastic gradient boosting [11, 12]. Specifically, we use the XGboost library [7] to implement the XGBoost method. The GPR has an average F1 score of 0.6911 whereas the XGBoost has an average F1 score of 0.5776. The GPR has a descent F1 score for the sampling rate of 1% (0.5505) as compared to XGBoost (0.4074).

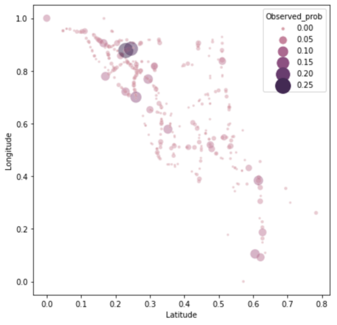
We also test the proposed framework by replacing logistic regression with XGBoost (column XGBoost-2 in Table 4). We observe that the XGBoost tends to overfit the training dataset and logistic regression gives better results for all four sampling rates in the test dataset.

We also applied the ABC algorithm to estimate the mean and standard deviation of each slice through the Python abcpy package [10]. Specifically, the parameter samples are generated from a uniform distribution ($\mu \sim (-0.5, 0.5)$, $\sigma \sim (0.5, 1.5)$) and a Gaussian likelihood is assumed for the distribution of the total flow. The posterior mean and standard deviation are then used to determine if a traffic flow is anomalous based on the three-standard deviation rule. We examine the hyperparameter ϵ (the distance tolerance) at two different values, 8 and 10, after comparing the simulation datasets and the observed data. We find the results for $\epsilon = 8$ and $\epsilon = 10$ are very close; therefore only the first one is shown in Table 4. It is shown that the proposed algorithm has higher F1 scores than the ABC algorithm for all four sampling rates.

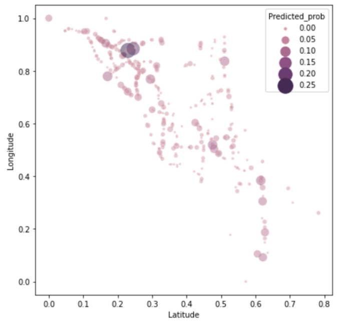
Furthermore, the F1 score on the test dataset using the baseline algorithm is also computed for comparison purposes. The baseline algorithm labels an observation as an anomaly if the detrended traffic flow is above or below three standard deviations from the observed mean. The proposed method has a higher F1 score for all four sampling rates than the baseline approach.

5.2 Spatial Pattern of Traffic Anomalies

In this section, we show the anomaly probability on a map using the standardized latitude and longitude information to reveal the spatial pattern of anomalies (i.e., the original latitude and longitude values are not provided). Due to the nature of the data, we aggregate the anomalies by year and compute both the observed and predicted anomaly probability for each sensor. The left and right plots in Figure 3 are the true and predicted anomaly probability in 2016, respectively. In general, the predicted anomaly probability is close to the observed anomaly probability. In both plots, it is shown that sensors with large anomaly probabilities are mainly located in the northwestern area which is likely close to downtown Los Angeles. We are not able to show it on the exact map since the latitude and longitude provided by NSF are normalized. Few sensors located in the southeastern area also have relatively high anomaly probabilities. The scatter plot also showed that sensors that are close to each other had similar anomaly probability.



(a) The observed anomaly probability for each sensor in year 2016



(b) The predicted anomaly probability for each sensor in year 2016

Figure 3: The left and right plots show the observed and predicted anomaly probability represented by the bubble size for each sensor in year 2016. The sensors that are close to each other have similar anomaly probabilities.

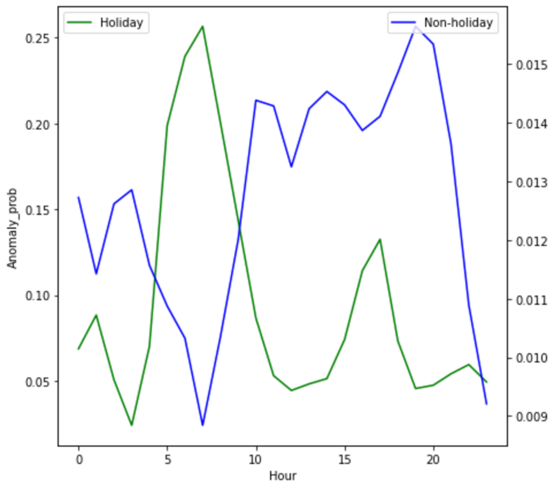


Figure 4: Hourly anomaly probability for holidays and non-holiday. The left vertical axis number is the probability of anomalies in holidays and the right vertical axis number is the estimated probability of anomalies in non-holidays.

5.3 Holiday and Weekday Pattern of Traffic Anomalies

In this section we examine the hourly traffic patterns of holidays and non-holidays and their daily profiles.

Holidays vs. non-holidays. We analyze the hourly anomaly probability to investigate holiday and weekday patterns (see Figure 4). In general, holidays have a higher probability of experiencing unusual traffic than non-holidays. Moreover, holidays generally had anomalous traffic in the morning (4 AM to 10 AM), whereas non-holidays had higher anomaly probabilities from 10 AM to 8 PM. For holidays, people tend to leave home early, so this may explain the peak of the anomaly probability in the early morning [6]. For non-holidays, the anomaly probability patterns coincide with daily work commute patterns. Figure 4 indicated that the daily profile had two peaks for non-holidays, corresponding to two rush hour periods: one in the morning (11 AM to 12 PM) corresponding to lunch breaks and another one in the afternoon (7 PM to 8 PM) which corresponds to the end of a working day.

Daily profiles in holidays and non-holidays. Stratifying the traffic flow data by holidays and non-holidays, the hourly anomaly probabilities are further analyzed at the weekday level. Figure 5 showed the daily profiles by days of the week for holidays and non-holidays. We observe that there are large traffic flows on holidays that occurred on Mondays through Saturdays from 7 AM to 8 PM whereas Sunday has large traffic flows at 1 AM and a lower anomaly probability overall. We also found that Tuesday, Wednesday and Thursday have higher a probability of experiencing anomalous traffic than other days in the holiday group. Also, we found that 1 AM has the second highest probability of

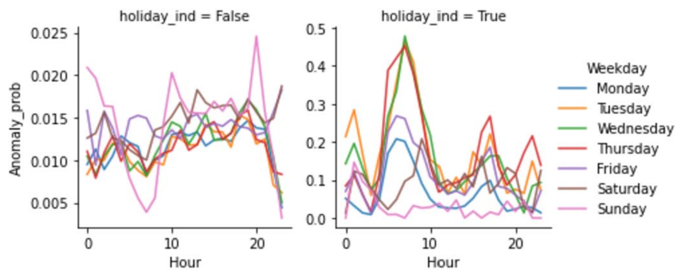


Figure 5: Hourly anomaly probability by weekday for holidays (right panel) and non-holidays (left panel). For holidays, all days have a high peak at 7 AM and 8 AM except for Sunday which have a high peak at 1 AM. For non-holidays, Monday, Tuesday, Wednesday, and Thursday have similar patterns, while Friday, Saturday and Sunday are similar.

experiencing anomalies for most holidays which is consistent with a previous study [15].

For non-holidays, we find that Monday, Tuesday, Wednesday, and Thursday have similar patterns: a) two peaks at 11 AM to 12 PM and 7 PM to 8 PM and b) fewer anomalies at 8 AM. Friday's pattern differes in that its anomaly probability gradually increases after 1 AM instead of dropping in the morning which is similar to Saturday and Sunday. It is interesting to point out that morning traffic volumes from Monday to Friday may be high but has a relatively smaller chance of reporting anomalies as compared to other time frames.

6. CONCLUSIONS

In the proposed algorithm, we introduce a data engineering process which consists of detrending, normalization based on weekdays and weekends, and data augmentation using temporal or spatial patterns. The data engineering step plays an important role in the algorithm and can be generalized to settings with missing data. The GPR works on data with low and high signal-to-noise ratios and can be scaled to very large traffic flow datasets using a straightforward, practical, and generally applicable model specification. It retains the temporal correlation and helps estimate the true mean and standard deviation for each slice of data together with linear regression models. The logistic regression model predicts anomaly probabilities for each slice of data, which can be used for improving traffic and road safety by roadway designers and traffic management departments.

There are several potential areas for future work. Although the Python scikit-learn package [18] optimizes the hyperparameters in GPR, other optimization strategies such as Bayesian optimization may be implemented to recover the complete data and approximate the mean and standard deviation [24]. Future work might also consider spatial correlation in GPR. For example, it can be extended to accommodate high-dimensional geostatistic data by using reduced rank and hierarchical nearest-neighbor Gaussian process models [3, 9]. The model could also be extended to a deep learning model by using GP as a hidden layer [28].

FUNDING

This research was supported in part by the National Science Foundation grant, DMS-1924792.

CONFLICTS OF INTEREST

The authors declare no conflict of interest. All authors reviewed the results and approved the final version of the manuscript.

Accepted 4 January 2023

REFERENCES

[1] Algorithms for threat detection (atd). URL. https://www.nsf.gov/pubs/2020/nsf20531/nsf20531.htm.

[2] Bai, S., He, Z., Lei, Y., Wu, W., Zhu, C., Sun, M. and Yan, J. Traffic anomaly detection via perspective map based on spatial-temporal information matrix. In *CVPR Workshops* 117–124 (2019).

[3] Banerjee, A., Dunson, D. B. and Tokdar, S. T. Efficient gaussian process regression for large datasets. *Biometrika* **100**(1) 75–89 (2013). https://doi.org/10.1093/biomet/ass068. MR3034325

[4] Beaumont, M. A. Approximate bayesian computation in evolution and ecology. *Annual review of ecology, evolution, and systematics* 379–406 (2010). https://doi.org/10.1146/annurev-statistics-030718-105212. MR3939526

[5] Bhaskaran, K. and Smeeth, L. What is the difference between missing completely at random and missing at random? *International Journal of Epidemiology* **43**(4) 1336–1339 (2014).

[6] Calafate, C. T., Soler, D., Cano, J.-C. and Manzoni, P. Traffic management as a service: The traffic flow pattern classification problem. *Mathematical Problems in Engineering* (2015).

[7] Chen, T. and Guestrin, C. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16*, New York, NY, USA 785–794 (2016). ACM. http://doi.acm.org/10.1145/2939672.2939785. ISBN .

[8] Csilléry, K., Blum, M. G., Gaggiotti, O. E. and François, O. Approximate bayesian computation (abc) in practice. *Trends in Ecology & Evolution* **25**(7) 410–418 (2010).

[9] Datta, A., Banerjee, S., Finley, A. O. and Gelfand, A. E. Hierarchical nearest-neighbor gaussian process models for large geostatistical datasets. *Journal of the American Statistical Association* **111**(514) 800–812 (2016). https://doi.org/10.1080/01621459.2015.1044091. MR3538706

[10] Dutta, R., Schoengens, M., Onnela, J.-P. and Abcpy, A. M. A user-friendly, extensible, and parallel library for approximate Bayesian computation. In *Proceedings of the platform for advanced scientific computing conference* 1–9 (2017).

[11] Friedman, J., Hastie, T. and Tibshirani, R. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The Annals of Statistics* **28**(2) 337–407 (2000). https://doi.org/10.1214/aos/1016218223. MR1790002

[12] Friedman, J. H. Greedy function approximation: a gradient boosting machine. *Annals of Statistics* 1189–1232 (2001). https://doi.org/10.1214/aos/1013203451. MR1873328

[13] Kut, A. and Birant, D. Spatio-temporal outlier detection in large databases. *Journal of Computing and Information Technology* **14**(4) 291–297 (2006).

[14] Little, R. J. and Rubin, D. B. *Statistical analysis with missing data* **793**. John Wiley & Sons, (2019). https://doi.org/10.1002/9781119013563. MR1925014

[15] Mihaita, A.-S., Li, H. and Rizoiu, M.-A. Traffic congestion anomaly detection and prediction using deep learning (2020). arXiv preprint. arXiv:2006.13215.

[16] Münz, G., Li, S. and Carle, G. Traffic anomaly detection using k-means clustering. In *GI/ITG Workshop MMBnet* 13–14 (2007).

[17] Neal, R. M. Priors for infinite networks. In *Bayesian Learning for Neural Networks* 29–53. Springer, (1996).

[18] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Scikit-learn, E. D. Machine learning in Python. *Journal of Machine Learning Research* **12**. 2825–2830 (2011). MR2854348

[19] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. Scikit-learn: Machine learning in python. *the Journal of machine Learning research* **12**. 2825–2830 (2011). MR2854348

[20] Quinonero-Candela, J., Rasmussen, C. E. and Williams, C. K. Approximation methods for gaussian process regression. In *Large-scale kernel machines* 203–223. MIT Press, (2007).

[21] Rasmussen, C. E. Gaussian processes in machine learning. In *Summer school on machine learning* 63–71. Springer, (2003).

[22] Roberts, S., Osborne, M., Ebden, M., Reece, S., Gibson, N. and Aigrain, S. Gaussian processes for time-series modelling. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* **371**, 20110550 (1984). 2013. https://doi.org/10.1098/rsta.2011.0550. MR3005668.

[23] Schulz, E., Speekenbrink, M. and Krause, A. A tutorial on gaussian process regression: Modelling, exploring, and exploiting functions. *Journal of Mathematical Psychology* **85**. 1–16 (2018). https://doi.org/10.1016/j.jmp.2018.03.001. MR3852577.

[24] Snoek, J., Larochelle, H. and Adams, R. P. Practical Bayesian optimization of machine learning algorithms. *Advances in Neural Information Processing Systems* **25** (2012).

[25] Sofuoglu, S. E. and Gloss, S. A. Tensor-based anomaly detection in spatiotemporal urban traffic data. In *Signal Processing* 108370 (2021).

[26] Williams, C. K. Computing with infinite networks. *Advances in neural information processing Systems* 295–301 (1997).

[27] Wilson, A. G., Hu, Z., Salakhutdinov, R. and Xing, E. P. Deep kernel learning. In *Artificial intelligence and statistics* 370–378. PMLR, (2016).

[28] Wilson, A. G., Hu, Z., Salakhutdinov, R. R. and Xing, E. P. Stochastic variational deep kernel learning. *Advances in Neural Information Processing Systems* **29**. 2586–2594 (2016).

[29] Zhang, M., Li, T., Shi, H., Li, Y. and Hui, P. A decomposition approach for urban anomaly detection across spatiotemporal data. In *IJCAI International Joint Conference on Artificial Intelligence. International Joint Conferences on Artificial Intelligence* (2019).

[30] Zhang, Z., He, Q., Tong, H., Gou, J. and Li, X. Spatial-temporal traffic flow pattern identification and anomaly detection with dictionary-based compression theory in a large-scale urban network. *Transportation Research Part C: Emerging Technologies* **71**. 284–302 (2016).

Qing He. Department of Statistics and Data Science, University of Central Florida, Orlando, Florida, USA.
E-mail address: carsonqing@knights.ucf.edu

Charles W. Harrison. Department of Statistics and Data Science, University of Central Florida, Orlando, Florida, USA.
E-mail address: charleswharrison@knights.ucf.edu

Hsin-Hsiung Huang. Department of Statistics and Data Science, University of Central Florida, Orlando, Florida, USA.
E-mail address: hsin.huang@ucf.edu