Detection and Characterization of Anomalies in Multivariate Time Series

Haibin Cheng, Pang-Ning Tan
Michigan State University
Lansing,MI,48823
chenghai,ptan@cse.msu.edu

Christopher Potter
NASA Ames Research Center
Moffett Field,CA
cpotter@mail.arc.nasa.gov

Steven Klooster
California State University
Monterey Bay,CA
klooster@gaia.arc.nasa.gov

Abstract

Anomaly detection in multivariate time series is an important data mining task with applications to ecosystem modeling, network traffic monitoring, medical diagnosis, and other domains. This paper presents a robust algorithm for detecting anomalies in noisy multivariate time series data by employing a kernel matrix alignment method to capture the dependence relationships among variables in the time series. Anomalies are found by performing a random walk traversal on the graph induced by the aligned kernel matrix. We show that the algorithm is flexible enough to handle different types of time series anomalies including subsequence-based and local anomalies. Our framework can also be used to characterize the anomalies found in a target time series in terms of the anomalies present in other time series. We have performed extensive experiments to empirically demonstrate the effectiveness of our algorithm. A case study is also presented to illustrate the ability of the algorithm to detect ecosystem disturbances in Earth science data.

1 Introduction

Anomaly detection is a valuable data mining technique for discovering unusual patterns in time series data. It can be used to detect interesting events such as heart arrythmia in electrocardiogram, intrusions in network data, congestion in traffic data, and ecosystem disturbances in Earth science data. While numerous algorithms have been developed for detecting anomalies [11, 15, 2, 12, 23, 24], most of them are designed for univariate time series.

The detection of anomalies in multivariate time series is more challenging for several reasons. First, it is difficult to establish a concise definition of an anomaly. Analogous to univariate time series, some anomalies may correspond to abnormally high (or low) values or unusual subsequences (discord [11]) in one or more time series. In addition, the multivariate anomalies may correspond to unexpected changes in the relationships among a set of variables [4]. For example, the time series for vegetation cover at mid-latitude locations in the United States typically varies in a 12-month cycle, peaking during the warm summer months and dropping to its minimum during the cold winter. Ecosystem disturbances such as wildfire and drought can be potentially detected based on the unusually low values of vegetation cover observed in the summer. Such anomalies are considered to be "local" (as opposed to global anomalies) since their values are abnormally low when compared to the average values observed during the warm summer months. Second, the performance of a multivariate anomaly detection algorithm is highly susceptible to the presence of noise in one or more time series. Therefore, a multivariate anomaly detection algorithm must be robust to noisy measurements in the time series data in order to increase its detection rate and to reduce its false alarm rate.

Multivariate time series is also useful to characterize the different types of anomalies found in a target time series. For example, Earth scientists are interested in detecting wildfires by monitoring the anomalies found in vegetation cover data derived from NASA's Earth observing satellites. However, analyzing the vegetation cover data alone is insufficient to distinguish between man-made wildfires from those induced by extreme climate events (such as lightning strikes from severe thunderstorms). Therefore, a key challenge is to combine the time series from other related variables (e.g., temperature and precipitation) to help explain the anomalies found in a target time series (vegetation cover).

To address these challenges, we develop a robust graph-based algorithm for detecting anomalies in multivariate time series data. Our algorithm learns the dependence relationships among variables in the multivariate time series by employing a kernel matrix alignment method. We empirically show that such alignment helps to eliminate noise and retains only anomalies in the target time series that can be explained by anomalies observed in other time series. The time series anomalies are detected by performing a random walk traversal on the graph induced by the aligned kernel matrix. In principle, a kernel matrix can be constructed from either a time point or a subsequence in the time series. Thus, our algorithm is flexible enough to handle different types of time series anomalies including subsequence-based and local anomalies. We demonstrate the effectiveness of our algorithm by using a number of synthetic and real data sets. A case study is also presented to illustrate the ability of the algorithm to detect ecosystem disturbances in Earth science data.

The remainder of this paper is organized as follows. Section 2 presents the preliminary background of this work. The graph-based algorithm for detecting anomalies is described in Section 3. Section 4 introduces our kernel alignment framework. Application of the framework to various time series anomaly detection problems is given in Sections 5 and 6. Section 7 reviews some of the previous work on time series anomaly detection. The performance of the proposed algorithms are examined in Section 8. Finally, we conclude with a summary of the work and directions for future research.

2 Preliminaries

Consider a time series $\boldsymbol{X} = x_1 x_2 \cdots x_T$, which is an ordered sequence of real-valued measurements taken at timestamps $1, 2, \cdots, T$. A multivariate time series $\mathcal{D} = \{\boldsymbol{X}_i\}_{i=1}^{p}$ is a collection of time series that corresponds to the measurements of p real-valued variables spanning the same time interval. For some application domains, one of the variables $Y \in \mathcal{D}$ may be designated as the target variable of interest, while the remaining $\mathcal{D} - \{Y\}$ variables are used as predictor variables.

2.1 Multivariate Time Series Anomaly Detection

The goal of anomaly detection in multivariate time series is to discover the timestamps at which the measurement values for one or more variables in $\mathcal{D}$ deviate significantly from their normal behavior. The normal behavior represents the expected value of a time series based on its historical changes as well as its relationship to other time series in $\mathcal{D}$. In this study, we consider two types of multivariate time series anomaly detection problems: *general* and *target specific*.

For general multivariate time series anomaly detection, all variables in $\mathcal{D}$ are considered equally important when detecting anomalies. For example, in network intrusion detection, anomalies can be detected based on the deviations observed in one or more time series—e.g., the unusually high number of connections originating from the same IP address (for a denial of service attack), the wide range of port numbers used (for a port scan attack), or the abnormally large number of ICMP packets sent (for a ping flood attack). In contrast, target specific anomaly detection aims to find anomalies in the time series for a target variable that are correlated with the deviations observed in other predictor time series. An example application can be found in the Earth science domain, where scientists are interested in identifying ecosystem disturbances such as wildfires from satellite observations of the global vegetation cover data (i.e., the target variable). By correlating the anomalies found in the vegetation cover data with those found in climate variables such as temperature and precipitation, this may help scientists to distinguish between climate-induced anomalies from human-induced anomalies.

Multivariate time series anomaly detection can also be categorized based on the type of anomaly found. For *pointwise anomaly detection*, the objective is to discover the timestamps at which the observed values are significantly different than the rest of the time series. For *subsequence anomaly detection*, the objective is to discover a segment of length $d \ll T$ that does not match any other segments in the time series. An example of such type of anomaly is shown in Figure 4(a). In both cases, the anomalies can be characterized as *global* or *local* anomalies, depending on how the normal behavior is determined. The normal behavior for global anomalies is computed from the entire time series whereas for local anomalies, the normal behavior is defined with respect to some local neighborhood information, including:

- **Time-based Neighborhood**: A data point at time t is considered a (time-based) local anomaly if its value is significantly different than the values observed within the time interval $[t - k, t + k]$. An example of such anomaly is shown in Figure 5(a).

- **Cycle-based Neighborhood**: Let k be the period of a time series. A data point at time t is considered a (cycle-based) local anomaly if its value is significantly different than the values observed at times $t - k, t - 2k, \cdots$ and $t + k, t + 2k, \cdots$.

2.2 Similarity Measure

A key element of an anomaly detection algorithm is the similarity measure used to determine how closely matched are two given observations. Let i and j denote a pair of timestamps in the time series. Each timestamp is associated with a set of observations for p variables. Although there are numerous similarity measures proposed in the literature (e.g., cosine, Jaccard, and correlation), few of them are applicable to both univariate and multivariate time series. In this study, we measure the similarity between two timestamps using the RBF function:

$$\boldsymbol{K}(i,j) = \exp\left[-\frac{\sum_{k=1}^{p}(x_{ki} - x_{kj})^2}{\sigma^2}\right], \tag{2.1}$$

where $p = 1$ for univariate time series. The RBF function can also be generalized to measure the similarity between two subsequences of length d as follows:

$$\boldsymbol{K}(i,j) = \exp\left[-\frac{\sum_{k=1}^{p}\sum_{s=0}^{d-1}(x_{k,i+l} - x_{k,j+l})^2}{\sigma^2}\right], \tag{2.2}$$

where i and j are the starting timestamps for each subsequence. The RBF function can be used to construct a non-negative, symmetric matrix $\boldsymbol{K}$, also known as a kernel matrix, that captures the pairwise similarity between every pair of timestamps (or subsequences) in a given time series.

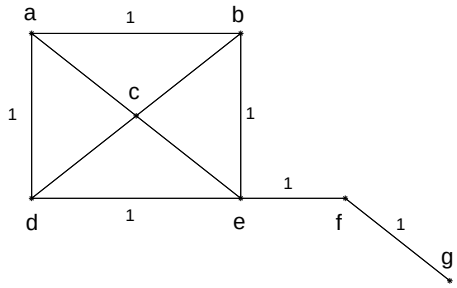


Figure 1: A weighted graph representation of a kernel matrix.

The next section describes a graph-based algorithm to detect anomalies based on similarity information contained in a kernel matrix.

3 Graph-based Anomaly Detection

A kernel matrix can be transformed into a weighted graph representation, $\mathcal{G} = (V, E)$, where V is the set of nodes and $E \subseteq V \times V$ is the set of edges. Each node in the graph corresponds to a data point (or a subsequence) in the time series while each weighted edge corresponds to the similarity value encoded in the kernel matrix $\boldsymbol{K}$. Because an anomalous node is highly dissimilar to other nodes, the weights of its connections to the rest of the nodes in the graph are generally small. Such a node will be rarely visited when performing a random walk traversal on the graph. This is the intuition behind our graph-based anomaly detection algorithm. For example, node g in Figure 1 is anomalous due to its low connectivity compared to other nodes in the graph.

To estimate the connectivity value of a node, we model the graph as a Markov chain on the state space V with a transition matrix S, whose $(i,j)^{\text{th}}$ element denote the transition probability from node i to node j. The transition matrix is obtained by normalizing each column of the kernel matrix.

$$S(i,j) = \frac{K(i,j)}{\sum_{i=1}^{n} K(i,j)} \tag{3.3}$$

The connectivity value of each node is computed iteratively by applying the following recursive equation:

$$\boldsymbol{c} = d/n + (1-d)S\boldsymbol{c}, \tag{3.4}$$

where d is the damping factor, n is the number of nodes, and $\boldsymbol{c}$ is a connectivity vector for all the nodes in the graph. This iterative procedure can be viewed as performing a random walk on the Markov chain, where given the current node u, there is a probability $1-d$ of visiting one of its neighboring nodes according to the transition matrix S and a probability d of visiting any random node in the graph. This approach is equivalent to the the formulation used by the PageRank algorithm [17]. Upon convergence, nodes with high connectivity values are considered normal whereas those with low connectivity values are declared as anomalous. As an example, Table 1 shows the connectivity values of the nodes given in Figure 1. Node g is detected as an anomaly because it has the lowest connectivity value.

a	b	c	d	e	f	g
0.138	0.139	0.178	0.139	0.191	0.132	**0.081**

Table 1: Connectivity values obtained by applying random walk algorithm on the graph shown in Figure 1. The lower the connectivity value, the more anomalous a node is.

4 Multivariate Kernel Alignment

As previously noted in Section 2.1, the normal behavior of a multivariate time series depends on the historical evolution of each time series as well as their relationships with each other. A kernel matrix captures only the pairwise similarity between observations at different timestamps but not the relationship between different time series. To apply the graph-based algorithm described in the previous section to multivariate time series, the kernel matrix must be adjusted to learn the dependence relationships among the time series in $\mathcal{D}$. This is accomplished by "aligning" their corresponding kernel matrices.

We begin with a discussion of how kernel alignment works when one of the time series corresponds to a target variable of interest. An extension of the framework to general multivariate time series problems, where all time series are equally important, is given in Section 5. Let $\boldsymbol{K_X}$ be the initial kernel matrix constructed from the set of predictor variables $\boldsymbol{X} \in \mathcal{D}$ (using Equation (2.1)) and $\boldsymbol{K}_Y$ be the corresponding kernel matrix computed from the target time series. The objective of kernel alignment is to derive an adjusted kernel matrix $\widehat{\boldsymbol{K}}_\alpha$ from $\boldsymbol{K_X}$ that maximizes its correlation to the target kernel matrix $\boldsymbol{K}_Y$, i.e.:

$$\max_{\boldsymbol{\alpha}} \frac{< \widehat{\boldsymbol{K}}_\alpha, \boldsymbol{K}_Y >_F}{\sqrt{< \widehat{\boldsymbol{K}}_\alpha, \widehat{\boldsymbol{K}}_\alpha >_F < \boldsymbol{K}_Y, \boldsymbol{K}_Y >_F}} \tag{4.5}$$

where $< \boldsymbol{A}, \boldsymbol{B} >_F = \sum_{ij} A_{ij} B_{ij}$ is the Frobenius inner product between two matrices. The aligned matrix $\widehat{\boldsymbol{K}}_\alpha$ is obtained by first decomposing $\boldsymbol{K_X}$ into a set of basis vectors $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_n\}$ and learning the corresponding weights α_i associated with each basis vector:

$$\max_{\boldsymbol{\alpha}} \frac{< \sum_{i=1}^{p} \alpha_i \boldsymbol{v}_i \boldsymbol{v}_i', \boldsymbol{K}_Y >_F}{m\sqrt{< \sum_{i=1}^{p} \alpha_i \boldsymbol{v}_i \boldsymbol{v}_i', \sum_{i=1}^{p} \alpha_i \boldsymbol{v}_i \boldsymbol{v}_i' >_F}}, \tag{4.6}$$

where $m =< \boldsymbol{K}_Y, \boldsymbol{K}_Y >_F$.

We consider two ways to decompose $\boldsymbol{K}$ into its basis vectors. The first approach extracts the normalized eigenvectors of $\boldsymbol{K}$ by solving the eigenvalue equation $\boldsymbol{K}\boldsymbol{v} = \lambda\boldsymbol{v}$, subject to the constraint $\boldsymbol{v}_i'\boldsymbol{v}_j = 1$ if $i = j$ and zero otherwise. As a result, kernel alignment reduces to the following optimization problem:

$$\max_{\boldsymbol{\alpha}} \frac{\sum_{i=1}^{n} \alpha_i < \boldsymbol{v}_i\boldsymbol{v}_i', \boldsymbol{K}_Y >_F}{m\sqrt{\sum_{i=1}^{n} \alpha_i^2}} \tag{4.7}$$

or equivalently,

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{n} \alpha_i < \boldsymbol{v}_i\boldsymbol{v}_i', \boldsymbol{K}_Y >_F - \mu\left[\sum_{i=1}^{n} \alpha_i^2 - 1\right] \tag{4.8}$$

which has the following closed form solution [5]:

$$\alpha_i = \frac{< \boldsymbol{v}_i\boldsymbol{v}_i', \boldsymbol{K}_Y >_F}{\sqrt{\sum_i < \boldsymbol{v}_i\boldsymbol{v}_i', \boldsymbol{K}_Y >_F^2}} \tag{4.9}$$

One potential limitation of this formulation is that $\widehat{\boldsymbol{K}}_\alpha$ tends to overfit $\boldsymbol{K}_Y$ and thus loses its similarity to the original kernel matrix $\boldsymbol{K}_{\boldsymbol{X}}$. To alleviate this problem, we propose a new objective function:

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{n} \alpha_i < \boldsymbol{v}_i\boldsymbol{v}_i', \boldsymbol{K}_Y >_F - \mu\left[\sum_{i=1}^{n} (\alpha_i - \lambda_i)^2\right] \tag{4.10}$$

where λ_i and $\boldsymbol{v}_i$ are the corresponding eigenvalues and eigenvectors of the kernel matrix $\boldsymbol{K}_{\boldsymbol{X}}$. With this modification, the aligned kernel matrix $\widehat{\boldsymbol{K}}_\alpha$ will preserve as much information in $\boldsymbol{K}$ as possible. The weight parameters α_i can be computed in closed form as follows:

$$\alpha_i = \lambda_i + \frac{< \boldsymbol{v}_i\boldsymbol{v}_i', \boldsymbol{K}_Y >_F}{2\mu} \tag{4.11}$$

The preceding equation shows that α_i increases when there is strong positive correlation between $\boldsymbol{v}_i\boldsymbol{v}_i'$ and the target kernel $\boldsymbol{K}_Y$. Furthermore, if $\mu \to \infty$, $\alpha_i \to \lambda_i$, which reduces $\widehat{\boldsymbol{K}_\alpha}$ back to the original kernel matrix $\boldsymbol{K}_{\boldsymbol{X}}$.

An alternative approach to using eigenvectors is to simply replace $\boldsymbol{v}$ by the time series for each predictor variable, i.e., $\boldsymbol{v}_i = \boldsymbol{X}_i$. In this case, the objective function to be maximized is

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{p} \alpha_i < \boldsymbol{X}_i\boldsymbol{X}_i', \boldsymbol{K}_Y >_F - \mu\left[\sum_{i=1}^{p}\sum_{j=1}^{p} \alpha_i\alpha_j < \boldsymbol{X}_i\boldsymbol{X}_i', \boldsymbol{X}_j\boldsymbol{X}_j' >_F - 1\right] \tag{4.12}$$

For brevity, we choose $\mu = 1$ for our experiments. Taking the derivative of the objective function with respect to α_i and setting it to zero, we obtain:

$$\begin{aligned} &< \boldsymbol{X}_i\boldsymbol{X}_i', \boldsymbol{K}_Y >_F \\ &= \sum_{j=1}^{p} \alpha_j < \boldsymbol{X}_i\boldsymbol{X}_i', \boldsymbol{X}_j\boldsymbol{X}_j' >_F \end{aligned} \tag{4.13}$$

The weight parameters α_j are estimated by solving a system of linear equations. Since α_j is associated with the original variable $\boldsymbol{X}_j$, a large α_j indicates a strong correlation between $\boldsymbol{X}_j$ and the target variable $\boldsymbol{Y}$.

Once the weight parameters have been estimated, the aligned kernel matrix is constructed as follows:

$$\widehat{\boldsymbol{K}}_\alpha = \sum \alpha_i\boldsymbol{v}_i\boldsymbol{v}_i'. \tag{4.14}$$

One potential complication from using an eigenvector-based approach to define the basis set $\{\boldsymbol{v}_i\}$ is that the elements of $\widehat{\boldsymbol{K}}_\alpha$ may become negative when the basis vectors contain negative values. In this situation, the aligned kernel can no longer be interpreted as a transition matrix for random walk traversal. To alleviate this problem, the elements of the matrix are shifted so that the minimum value of the matrix becomes zero, i.e., $\widehat{\boldsymbol{K}}_\alpha \to \widehat{\boldsymbol{K}}_\alpha + |\min(\widehat{\boldsymbol{K}}_\alpha)|$. This is equivalent to adding a constant weight to every edge in the graph. Another possibility is to use the original predictor variables as the basis vectors $\boldsymbol{v}_i$.

5 Multivariate Time Series Anomaly Detection Algorithms

This section presents our proposed multivariate time series anomaly detection algorithm. A high-level summary of the algorithm is shown in Algorithm 1. There are two variations to the proposed algorithm. The first variation is designed for target-specific anomaly detection whereas the second variation is designed to identify anomalies in general multivariate time series, where each variable is equally important. The difference between the two variations lies in the way the aligned kernel matrix, $\widehat{\boldsymbol{K}}_\alpha$, is computed.

Algorithm 1 Multivariate Time Series Anomaly Detection

Input: Multivariate time series $\mathcal{D}$.
Output: Connectivity vector $\boldsymbol{c}$.
Method:
1. $\widehat{\boldsymbol{K}}_\alpha \leftarrow$ KernelAlign($\mathcal{D}$)
2. $S \leftarrow$ Normalize($\widehat{\boldsymbol{K}}_\alpha$)
3. $\boldsymbol{c} \leftarrow$ RandomWalk(S)

For target specific anomaly detection, we first align the kernel matrix derived from the predictor variables with the kernel matrix of the target variable, as described in Section 4. To perform the alignment, the original kernel

matrix $\boldsymbol{K_X}$ is decomposed into its basis vectors before applying Equations (4.11) or (4.13) to determine the weight parameters α. We then construct a transition matrix S from the aligned kernel $\widehat{\boldsymbol{K}}$ by normalizing the columns of the matrix. Finally, a Markov chain random walk algorithm is applied to the graph induced from the aligned kernel matrix to obtain the connectivity values of the nodes (where each node represents either a timestamp or a segment of the time series). Since the connectivity values depend on the similarity measure, number of nodes, and other factors, we normalize the connectivity scores by subtracting their means and dividing their standard deviations. Nodes with connectivity scores less than a user-specified threshold are declared as anomalies. The advantages of applying kernel alignment before the random walk anomaly detection algorithm are:

- Kernel alignment identifies components in the predictor time series that are correlated with the target time series. As can be seen from Equations (4.11) and (4.14), the role of α is to transfer information from the target time series to the predictor time series. The more correlated a component v_i is to the target time series, the more significant it is in terms of determining the similarity between two time points (or two subsequences). Thus, the kernel alignment step can be viewed as learning a new weighted similarity matrix that takes into account the dependencies between the predictor and target time series.

- Any components of the predictor time series that are unrelated to the target time series will have lower weights. As a consequence, kernel alignment helps to reduce the impact of noise as well as anomalies found in the predictor time series that are independent of the target time series.

For general multivariate time series anomaly detection, where $\boldsymbol{X} = (\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_p)$, we consider each variable $\boldsymbol{X}_i$ in turn as a target and learn an aligned kernel matrix $\widehat{\boldsymbol{K}}_i$ between the target and all the remaining variables in $\boldsymbol{X}$. This produces p aligned kernel matrices. The overall aligned kernel matrix $\widehat{\boldsymbol{K}}_\alpha$ is obtained by taking the Hadamard product of the individually aligned kernel matrices $\widehat{\boldsymbol{K}}_i$:

$$\widehat{\boldsymbol{K}}_\alpha = \widehat{\boldsymbol{K}}_1 \circ \widehat{\boldsymbol{K}}_2 \cdots \circ \widehat{\boldsymbol{K}}_p \tag{5.15}$$

After computing $\widehat{\boldsymbol{K}}_\alpha$, the remaining steps are similar to those taken for target-specific anomaly detection. The main advantage of applying kernel alignment for the general multivariate time series anomaly detection is to reduce the impact of noise in one or more time series.

6 Extensions of the Proposed Algorithm

This section describes two extensions of the proposed algorithms: (1) to detect local anomalies in the time series, and (2) to discover subsequence-based anomalies.

6.1 Sparse Kernel for Local Anomaly Detection Our aligned kernel matrix is used to construct a graph $\mathcal{G}$ upon which a random walk algorithm is applied to identify anomalous nodes in the graph. In general, $\mathcal{G}$ is a fully connected graph whose edge weights depend on the values of the aligned kernel matrix $\widehat{\boldsymbol{K}}_\alpha$. As a result, the previous algorithm discovers global anomalies whose values are significantly different than the rest of the time series. The proposed algorithm can be extended to local anomalies by sparsifying the aligned kernel matrix prior to constructing the transition matrix S.

As mentioned in Section 2.1, we have implemented two ways to define the neighborhood of a node for local anomaly detection in time series data. The first approach, time-based neighborhood, is implemented by removing the edges between all pairs of nodes i and j in which $|i-j| > k$. Thus, a local anomaly observed at time t has significantly different values than other observations within the time interval $[t-k, t+k]$. For the second approach, i.e., cycle-based neighborhood, we sparsify the graph by removing all edges in $\mathcal{G}$ in which $|i-j| \mod k > \tau$. For example, setting $\tau = 0$ would compare an observation at time t to other observations at times $t-k, t-2k, \cdots$ and $t+k, t+2k, \cdots$, where k is the known periodicity of the time series.

6.2 Subsequence Anomaly Detection In addition to point-wise anomalies, our algorithm can also be extended to discover subsequence anomalies (also known as discords). Let $\boldsymbol{Y}$ be the target variable and $\boldsymbol{X}$ be the set of predictor variables. Assume T is the length of all the time series. A sliding window of predefined size d is used to extract all the subsequences of length d from the multivariate time series. The corresponding predictor and target variable subsequences are represented as $\overline{\boldsymbol{X}}$ and $\overline{\boldsymbol{Y}}$, respectively. The number of subsequence windows created from the time series data is $T-d+1$. Each element of $\overline{\boldsymbol{X}}$ and $\overline{\boldsymbol{Y}}$ is a subsequence with length d, which is denoted as:

$$\begin{aligned} \overline{x}_{ij} &= \{x_{i,j}, x_{i,j+1}, \ldots, x_{i,j+d-1}\} \\ \overline{y}_j &= \{y_j, y_{j+1}, \ldots, y_{j+d-1}\} \end{aligned}$$

The kernel matrices $\boldsymbol{K}$ and $\boldsymbol{K}_Y$ are constructed by applying the following Equations:

$$\begin{aligned} \boldsymbol{K}(i,j) &= \exp\left(-\frac{\sum_{k=1}^{p} ||\overline{x}_{ki} - \overline{x}_{kj}||^2}{\sigma^2}\right) \\ &= \exp\left(-\frac{\sum_{k=1}^{p}\sum_{l=0}^{d-1}(x_{k,i+l} - x_{k,j+l})^2}{\sigma^2}\right) \end{aligned}$$

and

$$\boldsymbol{K}_Y(i,j) \quad = \quad \overline{y}_i \overline{y}'_j = \sum_{l=0}^{d-1} y_{i+l} y_{j+l}$$

The (i,j)th element in the kernel matrix represents the similarity between two subsequences (i.e., sliding windows) of length d starting from the timestamps i and j. Similar to the approach described for point-wise anomaly detection, the kernel matrices can be aligned before applying the random walk algorithm to detect the anomalous subsequences. While the approach presented in this section assumes a fixed window size d, it can be easily extended to deal with variable length time windows as well as other similarity measures for subsequences.

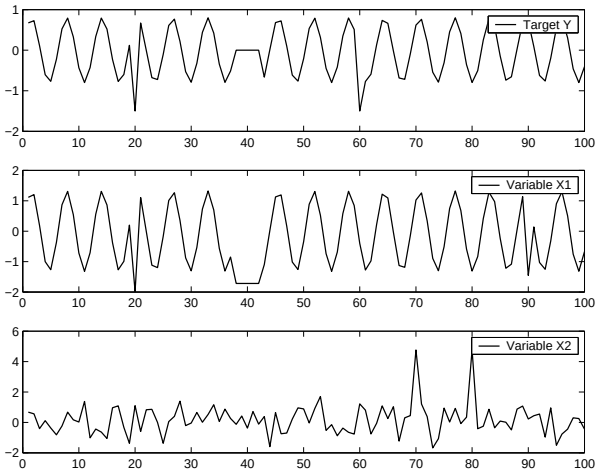
7 Related Work

Numerous algorithms have been proposed to discover unusual patterns in univariate time series. For example, Keogh et al. [11] developed an algorithm for mining time series that contain subsequences with largest nearest neighbor distance. Mahoney and Chan [15] employed a path and box feature trajectory approach to model the time series. Anomalous subsequences are detected based on their deviation from the trajectory path. Bay et al. [2] utilized local AR models to transform the data into its corresponding parameter space and detects anomalies based on distances computed in the parameter space. Other univariate approaches include frequency-based methods [12, 23], immunology-based method [6], and probability-based method [24].

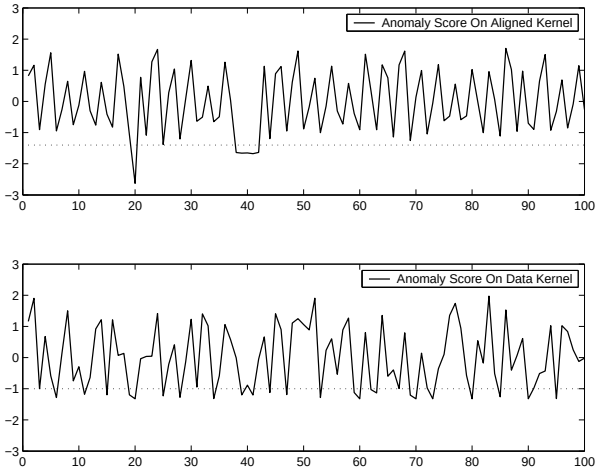
Recently there has been considerable interest in developing anomaly detection algorithms for multivariate time series. One category of methods utilize time series projection [8] and independent component analysis [1] to convert the multivariate time series into univariate time series. One potential limitation of these methods is the loss of information incurred when projecting the data into 1-dimensional space. Despite the rich literature on time series anomaly detection, there are few works on characterizing the discovered anomalies. Lakhina et al. [13] proposed an approach to characterize anomalies in network traffic flows. Their work however simply performs a separate anomaly detection for each variable without focusing on any specific target variable. Potter et al. [18] used association analysis as a postprocessing step to relate extreme climate events with ecosystem disturbances.

8 Experimental Results

We have performed extensive experiments to evaluate the performance of our proposed algorithm. All the experiments were conducted on a Windows XP machine with 3.0GHz CPU and 1.0GB RAM.



(a) Three simulated time series, with a target variable Y and two predictor variables X_1 and X_2.



(b) Anomaly scores after applying random walk algorithm on the aligned (top) and unaligned (bottom) kernel matrices.

Figure 2: Simulated time series for target-specific anomaly detection.

8.1 Effectiveness of Target Specific Anomaly Detection

The objective of this experiment is to illustrate the effectiveness of applying kernel alignment for target specific anomaly detection in multivariate time series. We simulated three equal length time series as shown in Figure 2(a). The top diagram represents the target variable Y whereas the bottom two time series correspond to the predictor variables X_1 and X_2. Both X_1 and Y are generated by interjecting large amplitude pulses to a periodic sinusoidal time series. The time series for X_2 is generated by interjecting large amplitude pulses to a random time series generated from the normal distribution $N(t, 0.8)$. The diagram clearly shows that two of the anomalies in the target variable Y (at timestamps 20 and 40, respectively) can be explained by the anomalies

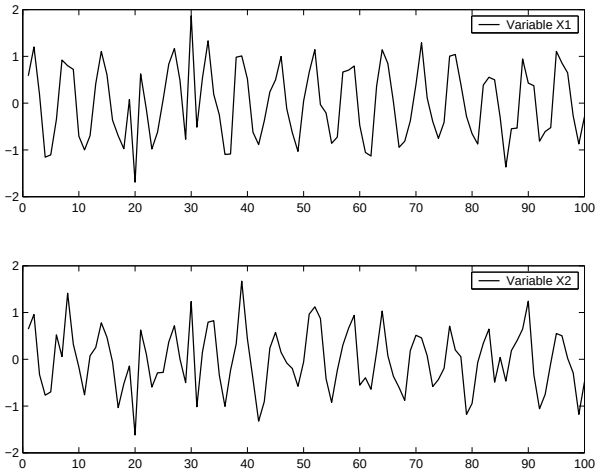
found in X_1. The third anomaly in the target variable Y (at timestamp 60) is unrelated to any anomalies observed in the predictor variables X_1 and X_2. Furthermore, the anomalies at timestamp 90 of the predictor variable X_1 and at timestamps 70 and 80 of the predictor variable X_2 do not correspond to any anomalies in Y.

Figure 2(b) shows the results of applying our target-specific anomaly detection algorithm on the aligned ($\widehat{\boldsymbol{K}}_\alpha$) and unaligned ($\boldsymbol{K}_{\boldsymbol{X}}$) kernel matrices derived from the combined predictor variables, X_1 and X_2. Anomalies are detected based on the timestamps at which the connectivity values are below certain threshold (after standardization). The results clearly demonstrate the effectiveness of applying kernel alignment to learn the dependence relationship between the predictor and target variables. Without kernel alignment, the correlated anomalies at timestamps 20 and 40 are indistinguishable from other timestamps in the time series, which may lead to high false positive or false negative rates (depending on the threshold chosen). By applying kernel alignment, anomalies that are correlated with those observed in Y (at timestamps 20 and 40) are amplified, which suggests that they can be used to characterize the anomalies found in the target variable. The anomalies found at timestamps 70 and 80 in the predictor variable X_1 and the anomaly found at timestamp 90 in predictor variable X_2 are degraded since they do not align with the target variable Y. Similarly, the anomaly at timestamp 60 in the target variable is also not detected because it cannot be explained by any anomalies in the predictor variables.

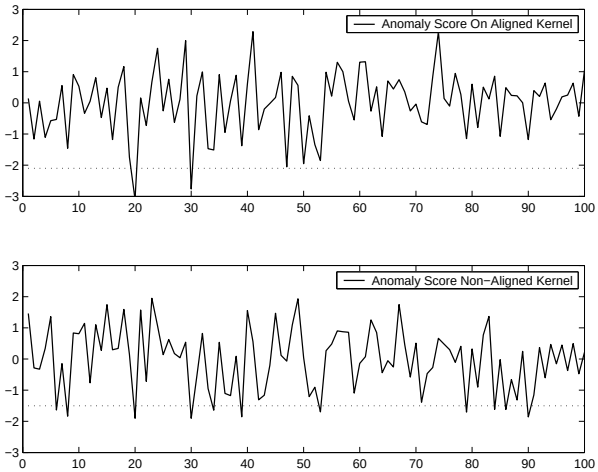
This experiment uses the original predictor time series as the basis vectors for the aligned kernel (see Section 4). The weights associated with the predictor variables are $\alpha_1 = 0.9964$ and $\alpha_2 = 0.0844$, respectively. These parameters suggest that X_1 is more correlated to the target variable Y than X_2, which is consistent with our observation.

8.2 Effectiveness of General Multivariate Time Series Anomaly Detection The purpose of this experiment is to demonstrate the effectiveness of applying kernel alignment on anomaly detection for multivariate time series without a target variable. Figure 3(a) shows two simulated time series X_1 and X_2 with a pair of anomalies at timestamps 20 and 30, respectively. A Gaussian noise ($N(0, 0.3)$) is also added to each time series to distort the signals.

The resulting anomaly score is plotted in Figure 3(b). Without kernel alignment, the random walk algorithm has difficulty in discriminating the anomalies at timestamps 20 and 30 from other noisy signals in the time series. By applying kernel alignment (using Equation (5.15)), the random walk algorithm can successfully detect both anomalies in the noisy time series. This experiment suggests that kernel alignment helps to improve the kernel matrix used by the anomaly detection algorithm by reducing the effect of noise.



(a) Two simulated noisy time series X_1 and X_2.

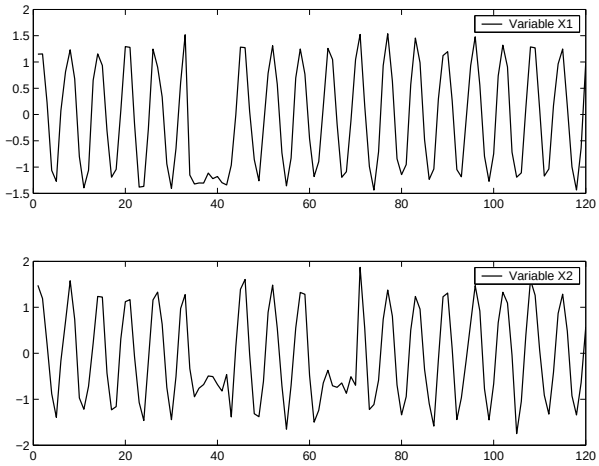


(b) Anomaly scores obtained by applying random walk algorithm on the aligned (top) and unaligned (bottom) kernel matrices.

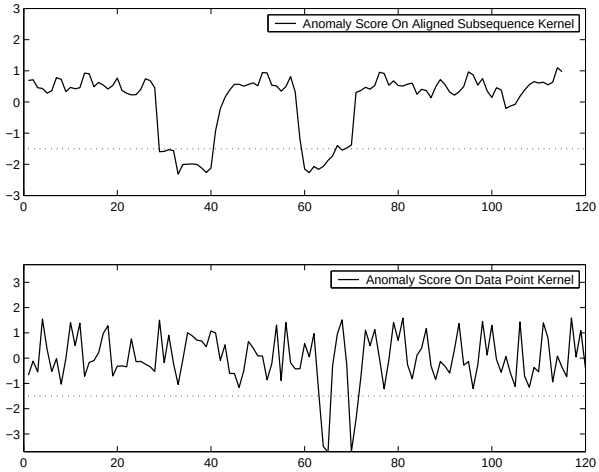
Figure 3: Simulated time series for general multivariate anomaly detection.

8.3 Subsequence Anomaly Detection The purpose of this experiment is to demonstrate the effectiveness of our algorithm for finding unusual subsequences in a multivariate time series. We simulated two time series X_1 and X_2 as shown in Figure 4(a). A Gaussian noise (with mean zero and standard deviation 0.1) is used to distort both time series. Anomalous subsequences are then added to both time series.

We compare the results of applying random walk anomaly detection algorithm on kernel matrices constructed from the subsequences to those constructed from individual time points. In both cases, the kernel matrices are initially aligned before applying the random walk algorithm. The subsequence-based kernel matrices are constructed using a sliding window of length 6. Figure 4(b) compares the anomaly scores obtained using the aligned subsequence-

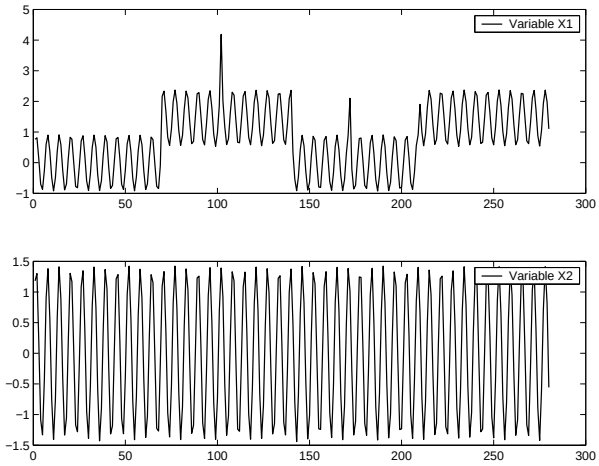


(a) Two simulated time series X_1 and X_2 with unusual subsequences.

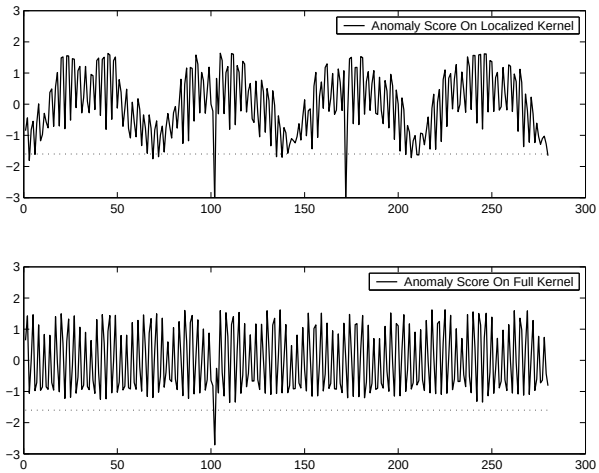


(b) Anomaly scores obtained by applying random walk algorithm on kernel matrices constructed from subsequences (top) and individual time points (bottom).

Figure 4: Detection of unusual subsequences.



(a) Two simulated time series X_1 and X_2 with local and global anomalies.



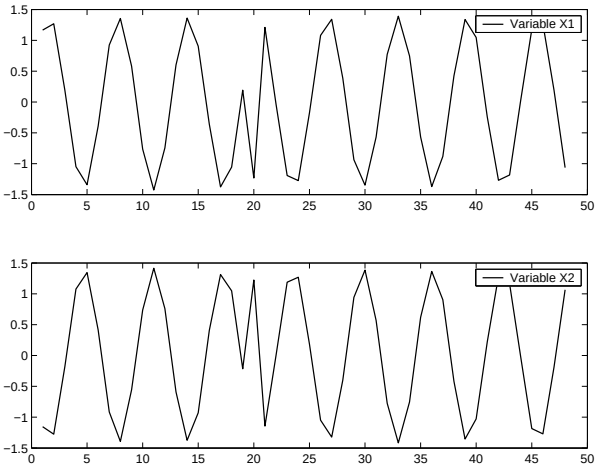
(b) Anomaly scores obtained by applying random walk algorithm on the time-based sparse (top) and full (bottom) kernel matrices.

Figure 5: Time-based local anomaly detection.

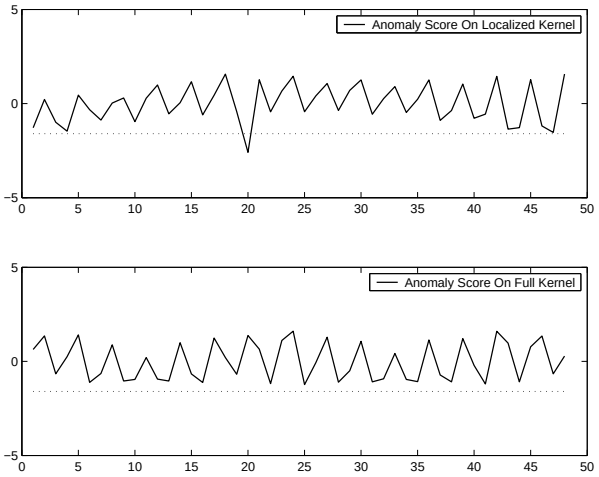
based kernel matrix (top diagram) to those obtained using the aligned point-wise kernel matrix (bottom diagram). The results obtained using subsequence-based kernel is clearly more superior.

8.4 Detection of Local Anomalies This experiment compares the results of applying the full versus sparse kernel matrices for detecting local anomalies. First, we illustrate the results of detecting time-based local anomalies. Figure 5(a) shows the two simulated time series used in our experiment. One of the time series, X_1, contains both a global anomaly (at timestamp 100) and a local anomaly (at timestamp 170). We apply the random walk anomaly detection algorithm on the graph induced by the sparse kernel matrix (using time-based neighborhood with $k = 20$), as described in Section 6.1. The resulting anomaly scores are plotted in Figure 5(b) and compared against those obtained by using the full (global) kernel. The results show that a full kernel may miss the local anomaly because its value at timestamp 170 is similar to other observations in X_1. In contrast, the sparse kernel is capable of detecting both global and local anomalies.

Next, we illustrate the results of detecting cycle-based local anomalies. The two simulated time series used for this experiment are shown in Figure 6(a). Both of the simulated time series contain a local anomaly at timestamp 20 and have a periodicity equals to 6. Clearly, the values of the variables at timestamp 20 is not unusual from a global perspective. As a result, standard distance-based and density-based anomaly detection methods fail to detect



(a) Two simulated time series X_1 and X_2 with cycle-based anomalies.



(b) Anomaly scores obtained by applying random walk algorithm on the cycle-based sparse (top) and full (bottom) kernel matrices.

Figure 6: Cycle-based local anomaly detection.

such anomalies. Our random walk algorithm using the full kernel matrix also fails as shown in the bottom panel of Figure 6(b). However, by sparsifying the kernel matrix based on the cycle-based neighborhood approach, the anomaly at timestamp 20 is successfully detected as shown in the top panel of Figure 6(b).

8.5 Performance Comparison In this experiment, we compare the performance of our general multivariate time series anomaly detection algorithm against several baseline algorithms using benchmark 2-D anomaly detection data[1] from the University of California Riverside time series data mining archive [10]. The benchmark data, which is used to

[1]The data was originally created by Dasgupta et al. [9] and obtained via a CD-Rom from Eamonn Keogh.

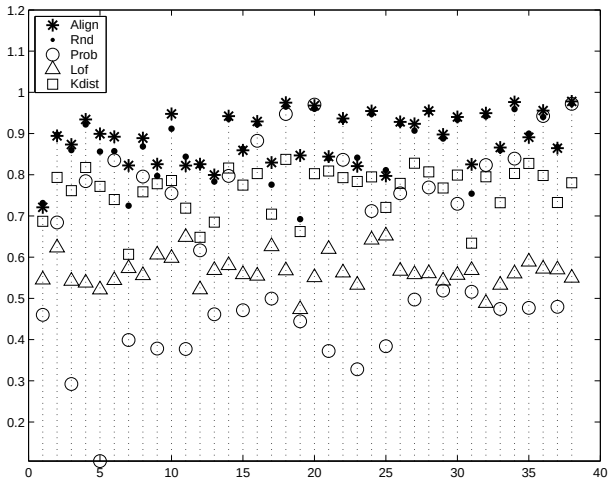


Figure 7: Performance comparison using 38 data sets from UC Riverside data repository.

test the performance of anomaly detection algorithms, contain 38 distinct time series data (including `Comb`, `Ring-thin`, `Stripe-mid`, `Cross-thick`, etc). Each time series data has a separate training and test sets. The training set contains a pair of "normal" time series of length 1000. The test set also has a pair of time series of length 1000 in addition to a class label vector that indicates whether a data point in the bivariate time series is anomalous.

We compared our general multivariate time series algorithm (denoted as `Align`) against 4 other competing methods—random walk without alignment [16], probability based [24], LOF [3], and K-distance based methods [19]. Note that the probability based method trains a classifier from the training set and thus is a supervised methods. All other methods, including `Align`, are unsupervised and do not use the training set. The parameters of the algorithms are determined as follows. The σ parameter for RBF function used in random walk (with and without alignment) is set to 0.02. For LOF and K-dist methods, we vary the number of nearest neighbors from 4 to 10. However, we notice that the results do not vary considerably. So, we set the number of nearest neighbors to be 6 in our experiments. The performance of the algorithms is measured in terms of their area under ROC curve (AUC) [7]. Figure 7 shows the AUC values for all the methods on the 38 data sets. The results suggest that `Align` significantly outperforms other existing methods for the majority of the data sets. As an example, Figure 8 shows the ROC curves for all the methods on one of the data sets in which our random walk algorithm on the aligned kernel matrix has the largest area.

8.6 Application to Ecosystem Disturbance Detection This section describes the results of applying our algorithm

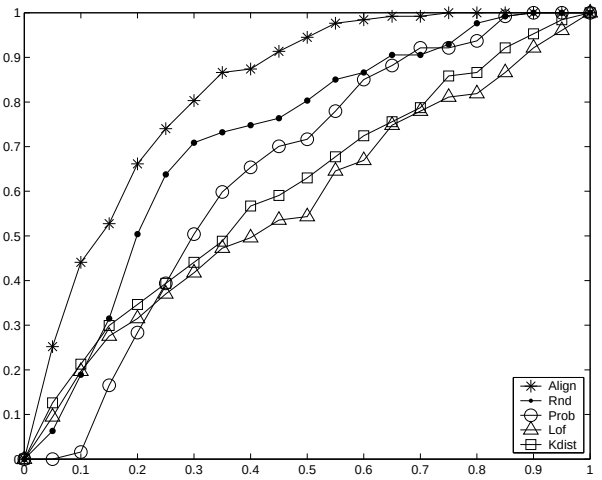
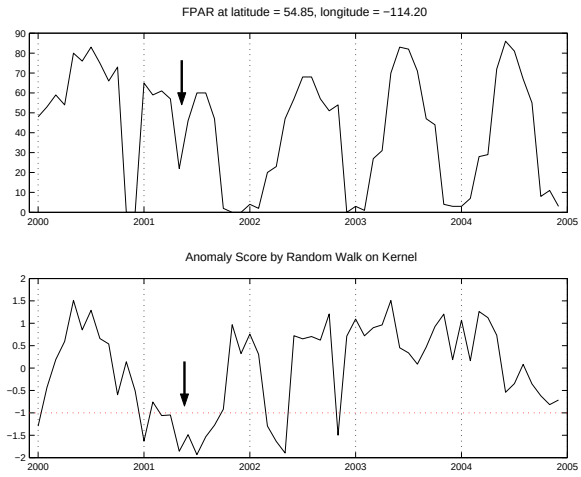


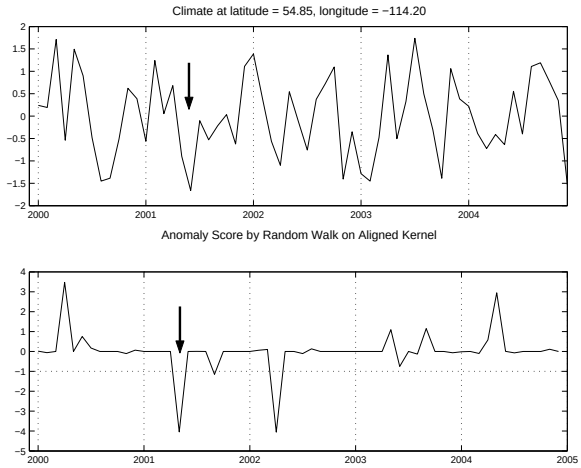
Figure 8: An example of ROC curves for all the methods on the 7_{th} data set.

to the problem of detecting ecosystem disturbances such as wildfires in Earth science data. Specifically, ecosystem disturbances are detected by monitoring changes in the vegetation cover data (called FPAR [21]) obtained from satellite observations. The monthly FPAR data is available at 4km $\times$ 4km spatial resolution covering the time period between 2000 and 2005. The objective of this study is to detect ecosystem disturbances using our proposed algorithm and to correlate them against the anomalies observed in climate data (such as precipitation, temperature, and sea-level pressure). The FPAR time series is treated as the target variable of interest, whereas the climate time series are used as the predictor variables.

To detect anomalies, we align the kernel matrix constructed from the climate variables against the kernel matrix for FPAR. The aligned kernel is then sparsified using the cycle-based neighborhood approach with $k = 12$ and $\tau = 1$. The anomaly scores for the multivariate time series are computed by performing random walk on the graph induced by the aligned kernel matrix. Using this approach, we were able to detect several incidents of large-scale FPAR disturbance events that correlate with extreme climate conditions. One example is given in Figure 9(a), which shows the FPAR time series at a location in Alberta, Canada (latitude=54.85N, longitude=114.20W) along with its anomaly scores obtained by applying the random walk algorithm to the kernel matrix constructed from the FPAR time series. The timing and location of the FPAR disturbance event coincide with the Chisholm wildfire event on May, 2001 as reported [20]. It has also been reported that dry conditions is one of the factors causing the severity of the wildfire. Figure 9(b) shows the monthly precipitation for the location along with the anomaly scores computed from the aligned



(a) FPAR time series and its anomaly score.



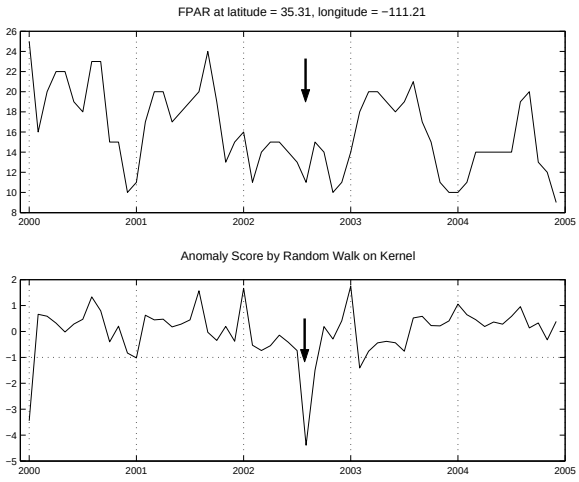
(b) Precipitation time series and its anomaly score.

Figure 9: FPAR and Precipitation time series at ($latitude = 54.85N, longitude = 114.20W$) and corresponding anomaly score by random walk.

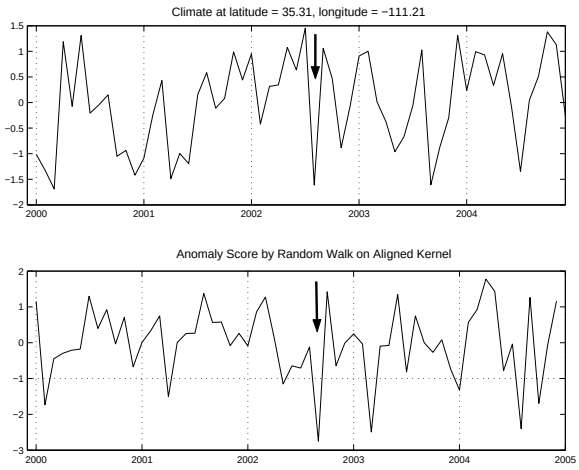
kernel. Observe that the FPAR disturbance event is well-aligned with the occurrence of a low precipitation event[2]

As another example, Figure 10(a) shows the FPAR time series and anomaly score for a location in South Dakota (latitude=35.31N, longitude=111.21W). The timing and location of the disturbance event co-incide with the Antelope wildfire [22] reported in August, 2002. The FPAR anomaly is also associated with a low precipitation event, as shown in Figure 10(b). Although there are other unusual precipitation events during this period, only one of them coincides with the FPAR disturbance.

[2]Our analysis also suggests the possibility of another FPAR disturbance and low precipitation event in Spring 2002. However, more analysis is needed to validate this event.

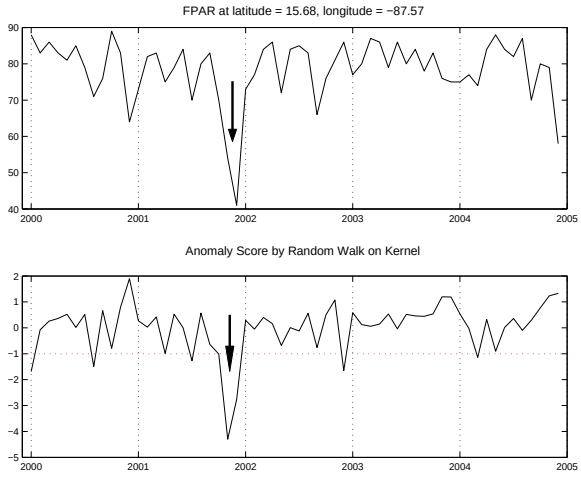


(a) FPAR time series and its anomaly score.

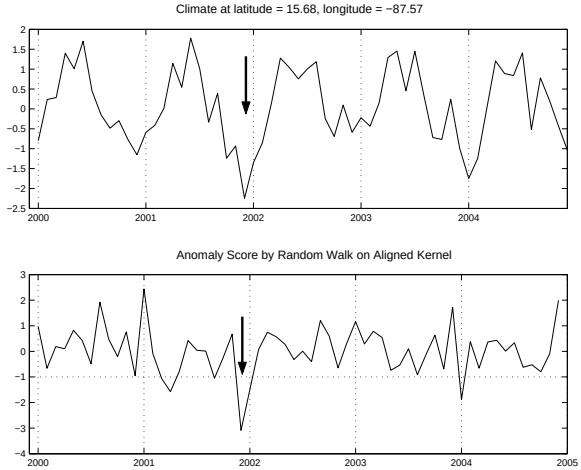


(b) Precipitation time series and its anomaly score.

Figure 10: FPAR and Precipitation time series at (latitude=35.31N, longitude=111.21W) and corresponding anomaly score by random walk.



(a) FPAR time series and its anomaly score.



(b) Sea level pressure and its anomaly score.

Figure 11: FPAR and Sea level pressure time series at (latitude=15.68N, longitude=87.57W) and corresponding anomaly score by random walk.

A third example is given in Figure 11(a) for the FPAR time series at a location in Honduras (latitude=15.68N, longitude=87.57W). A sudden drop in FPAR was observed at the end of 2001, which correlates with a sudden drop in sea level pressure (see Figure 11(b)). According to a report by the National Oceanic and Atmospheric Administration (NOAA) [14], the FPAR disturbance event coincides with the timing of Hurricane Iris.

9 Conclusions

Detecting and characterizing anomalies in multivariate time series is a important task with wide applications in different domains. In this work, we present a robust graph-based approach for multivariate time series anomaly detection. Our framework is very flexible and can be extended to detect unusual subsequences and local anomalies. We have conducted extensive experiments on both real and synthetic data sets to demonstrate the effectiveness of our algorithm. For future work, we propose to investigate the effectiveness of the proposed algorithms in the presence of concept drifts and missing values in the multivariate time series.

10 Acknowledgments

This work is supported by NSF grant #0712987. The authors would like to thank Dr Eamonn Keogh for providing them with the 38 time series data used in their experiments.

References

[1] R. Baragona and F. Battaglia. Outlier detection in multivari-

ate time series by Independent Component Analysis. *Neural Computation*, 19(1):1962–1984, January 2007.

[2] S. Bay, K. Saito, N. Ueda, and P. Langley. A framework for discovering anomalous regimes in multivariate time-series data with local models. Technical report, Center for the Study of Language and Information, Stanford University, 2004.

[3] M. Breunig, H. Kriegel, R. Ng, and J. Sander. LOF: identifying density-based local outliers. In *Proceedings of of the ACM SIGMOD International Conference on Management of Data*, pages 93–104, 2000.

[4] H. F. Chen, H. Cheng, G. F. Jiang, and K. J. Yoshihira. Exploiting local and global invariants for the management of large scale information systems. In *Proceedings of the 8th IEEE International Conference on Data Mining*, Pisa, Italy, 2008.

[5] N. Cristianini, J. Shawe-Taylor, A. Elisseeff, and J. Kandola. On Kernel Target Alignment. In *Advances in Neural Information Processing Systems 14*, pages 367–373, Vancouver, Canada, 2001.

[6] D. Dasgupta and S. Forrest. Novelty detection in time series data using ideas from immunology. In *Proceedings of the 5th International Conference on Intelligent Systems*, 1996.

[7] T. Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, June 2006.

[8] P. Galeano, D. Pena, and R. S. Tsay. Outlier detection in multivariate time series via projection pursuit. *Journal of the American Statistical Association*, 101(474):654–669, 2006.

[9] Z. Ji and D. Dasgupta. Applicability issues of the real-valued negative selection algorithms. In *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, pages 111–118, Seattle, WA, 2006.

[10] E. Keogh and T. Folias. UCR Time Series Data Mining Archive. `http://www.cs.ucr.edu/~eamonn/TSDMA/index.html`, 2002.

[11] E. Keogh, J. Lin, and A. Fu. HOT SAX: Efficiently Finding the Most Unusual Time Series Subsequence. In *Proceedings of the Fifth IEEE International Conference on Data Mining*, pages 226–233, 2005.

[12] E. Keogh, S. Lonardi, and B. Chiu. Finding surprising patterns in a time series database in linear time and space. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 550–556, 2002.

[13] A. Lakhina, M. Crovella, and C. Diot. Characterization of network-wide anomalies in traffic flows. In *Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement*, pages 201–206. ACM, 2004.

[14] M. B. Lawrence. Tropical cyclone report - tropical storm lorenzo,national hurricane center. `http://www.nhc.noaa.gov/2001iris.html`, 2001.

[15] M. Mahoney and P. Chan. Trajectory boundary modeling of time series for anomaly detection. In *Proceedings of 11th SIGKDD International Conference on Knowledge Discovery and Data Mining Workshop on Data Mining Methods for Anomaly Detection*, 2005.

[16] H. D. K. Moonesinghe and P. N. Tan. Outlier detection using random walks. *Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence*, pages 532–539, January 2006.

[17] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the web. Technical Report SIDL-WP-1999-0120, Stanford InfoLab, 1999.

[18] C. Potter, P. N. Tan, M. Steinbach, S. Klooster, V. Kumar, R. Myneni, and V. Genovese. Major disturbance events in terrestrial ecosystems detected using global satellite data sets. *Global Change Biology*, pages 1005–1021, 2003.

[19] S. Ramaswamy, R. Rastogi, and K. Shim. Efficient algorithms for mining outliers from large data sets. *SIGMOD Record*, 29(2):427–438, 2000.

[20] B. Stocks, J. A. Mason, J. B. Todd, E. Bosch, B. Wotton, B. Amiro, M. Flannigan, K. Hirsch, K. Logan, D. Martell, and W. Skinner. Large forest fires in canada. *Journal of Geophysical Research*, pages 1959–1997, 2002.

[21] Y. Tian, Y. Zhang, Y. Knyazikhin, R. B. Myneni, and S. W. Running. Prototyping of MODIS LAI/FPAR algorithm with LASUR and Landsat data. *IEEE Transaction on Geoscience and Remote Sensing*, pages 2387–2401, 2000.

[22] U. S. Geological Survey. Antelope Fire, Sioux Falls, South Dakota. `ftp://edcftp.cr.usgs.gov/pub/data/landcover/files/2002/wupa/ante02a\_meta.pdf`, August 2002.

[23] L. Wei, N. Kumar, V. N. Lolla, E. Keogh, and S. Lonardi. Assumption-free anomaly detection in time series. In *Proceedings of the 17th International Scientific and Statistical Database Management Conference*, pages 237–240, 2005.

[24] K. Yamanishi and J. Takeuchi. A unifying framework for detecting outliers and change points from non-stationary time series data. In *Proceedings of SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2002.