Correlating Heterogeneous Time Series using Change Information

ABSTRACT

Calculating the correlation between different time series is an important data mining task. By using correlation method, we can find hidden relationships between different time series. However, in most real-world problems, time series often have many different patterns (e.g. Periodical, Linear, random, etc.). And the correlation between these heterogeneous time series is also very meaningful. Despite their importance, there has been little previous work addressing the correlation between two types of heterogeneous time series data.

In this paper, we propose a change based correlation method that is capable of evaluating evaluate the correlation between heterogeneous time series. In addition, by taking the advantage of this correlation coefficient, we propose to use LSH method to dealing with very large top-k searching problems. The experimental results on Synthetic data sets and real-world data sets show the effectiveness and efficiency of our algorithms.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Data Mining

General Terms

Application

Keywords

Correlation, Time Series, Hashing Learning

1. INTRODUCTION

The focus of this paper will be on the time series data. A time series is defined as a sequence of values {} associated with timestamps .

Time series mining is ubiquitous in data driven applications including robotics, medicine [], speech [25], object detection in vision [34, 28], system failure diagnosis [16, 29], ... more from the paper. Due to its pervasive presence, time series mining have received significant attention in recent years. A common prerequisite for data mining algorithms, such as clustering, search, classification and regression, etc., is a measure of correlation (or similarity). Dynamic Time Warping (DTW) is widely accepted, and arguably the most popular, measure of similarity (or correlation) for time series data in general [].

All existing measures over time series, including the popular DTW, relies on the notion that two time series T_1 and T_2 are similar, if there is a long enough similarly behaving subsequence common between them. This is a very reasonable notion which is also the desired nature of the similarity function in many applications. However, there are plenty of real-world problems where the notions of similarity can be very different. Using existing similarity measures, such as DTW, for such problems often lead to misleading results. Despite their importance, there has been little previous work addressing those scenarios. To signify their importance, we provide two motivating real-world examples:

Analysis of ECG (Electrocardiogram) data

Electrocardiography [10] (ECG or EKG*) is the process of recording the electrical activity of the heart over a period of time using electrodes placed on a patient's body. These electrodes detect the tiny electrical changes on the skin that arise from the heart muscle depolarizing during each heartbeat. If two ECG often have tiny electrical changes at the same time, they can be regarded as correlated.[17]. By analyzing such data, one can find some useful information hidden behind the human body, thus to uncover some miracles of human body [31].

We take the ECG time series data from the UCR time series data Archive [4], and the label is also provided by them. We choose three time series from two different labels in that data set. We illustrate this in Figure1, where we show the hierarchical clustering of these three time series under various measures. Top two red bold time series ($ECG\#1, ECG\#2$) are labeled as same class, and $ECG\#3$ is labeled as other class.

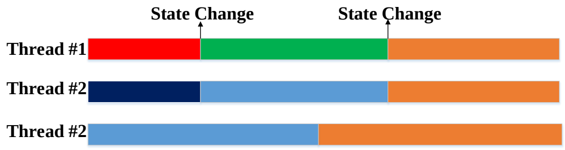

Figure 2: Three Thread time series

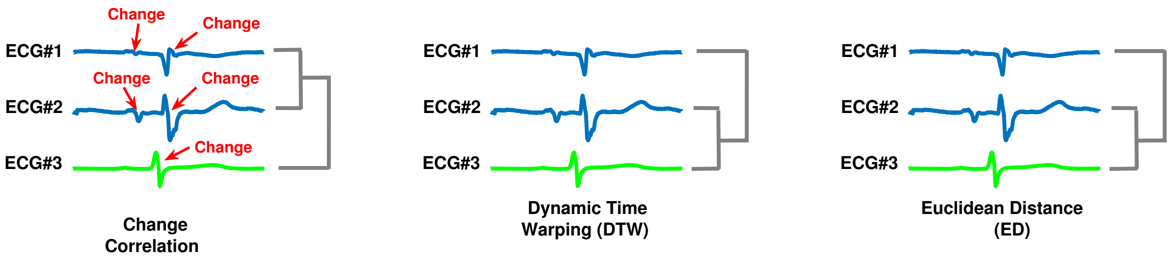

Figure 1: Three ECG time series with two labels

As we can see from Fig.1, Euclidean Distance and DTW Distance does poorly here. This is not surprising, since the ECG time series data have different change patterns (e.g. Increase-Decrease, Decrease-Increase, or a Sudden wave, etc.). These different change patterns can effect the performance of point to point similarity measures. And the correlation of ECG time series is major focus on whether have tiny electrical change at the same time [31].

Clear mention, where the data comes from, what are the labels (categories whyc you colored blue and red). And what is the value of the pairwise similarity between those 3 examples with DTW, our measure and L2 distance. Argue why DTW and L2 does not lead to the required ordering. Explain in details and we will condense it later. Right now its handwavy and wont work.

Analysis of HPC Thread time series. High-performance computers (HPC) have become enormously complex. Today, the largest systems consist of more than tens of thousands of nodes. Nodes themselves are equipped with one or more multicore microprocessors[1]. High-performance computer can generate over billions of threads during running. And how to automatically analysis and monitoring the HPC is a major task for HPC researchers [18, 30].

HPCToolkit[1] can be used to generate the performance information of each thread from HPC. And each thread can be represented as a time series. The value of the time series denote the state of the thread.

Fig. 2 shows a example of three thread time series data. Different colors means different state they are in. **Thread #1**, and **Thread #2** are in the same class, and **Thread #3** is in the other class. The ground truth is provided by the HPCToolkit team[2].

The DTW distance between **Thread #1**, and **Thread #2** is 0.83. While the DTW distance between **Thread #2**, and **Thread #3** is 0.31. While the DTW distance between **Thread #1**, and **Thread #3** is 0.51.

The Euclidean distance between **Thread #1**, and **Thread #2** is 0.64. While the DTW distance between **Thread #2**, and **Thread #3** is 0.53. While the DTW distance between **Thread #1**, and **Thread #3** is 0.71.

The Change Correlation between **Thread #1**, and **Thread #2** is 1.00. While the DTW distance between **Thread #2**, and **Thread #3** is 0.00. While the DTW distance between **Thread #1**, and **Thread #3** is 0.00.

We can see that DTW and Euclidean distance can not perform well. Because **Thread #1** and **Thread #2** always have different states and DTW and Euclidean will regard these state difference as large distance.

On the other hand, change based correlation only consider the change information of the time series, and state difference between **Thread #1**, and **Thread #2** can not effect the correlation result.

As showed in above examples, most of the existing point to point time series similarity measures (e.g. L1-Distance, L2-Distance [8], and DTW-Distance [21], etc) or correlation measures (e.g. Pearson Correlation [22], Kendall rank correlation [14], and Spearman's rank correlation [23], etc.) can not deal with such heterogeneous properties of the time series. The reason is: for heterogeneous time series, the correlation information is often associated with the change (During a period of time) of time series, rather than a point-to-point relationship in the traditional correlation analysis techniques. We will introduce the related research of point to point based similarity measure in detail in Section 5.

As a result, in order to deal with heterogeneity properties of time series, we proposed a change based correlation coefficient. The intuition of this correlation is: ***If two time series often change at the same time, they may have correlation with each other.*** . The mathematical definition of change based correlation is introduced in section 2.

Our change based correlation method firstly extract the change information of the time series data, and then use the change information to calculate the correlation coefficient between the two time series. By taking the advantage of this correlation coefficient, we propose to use LSH method to dealing with very large top-k searching problems.

The contribution of this paper is listed as follow:

1. Motivated by real applications, we investigate the correlation problem as between heterogeneous time series . To the best of our knowledge, this is the first attempt to evaluate the correlation between time series with different patterns.

2. We proposed a correlation coefficient between heterogeneous time series. By taking the advantage of this correlation coefficient, we propose to use LSH method to dealing with very large top-k searching problems.

3. The experiments on Synthetic data sets and Real world data sets show the effectiveness and efficiency of our method.

The rest of the paper is organized as follows: In Section 2, we introduce the problem statement and formulation. Our

[1]http://hpctoolkit.org/
[2]http://hpctoolkit.org/

approach is proposed in Section 3. The Empirical evaluation is shown in Section 4. In Section 5, we introduce some related works. Finally, we conclude our work in Section 6.

2. BACKGROUND

2.1 Preliminary Definition

In this section, we formally define some concept of this work, including Time Series, Change Point, Change Point set, Time Series Correlation.

DEFINITION 1 (TIME SERIES). *A time series, denoted as $S = (s_1, s_2, ..., s_m)$, where m is the number of points in the time series. The timestamps of a time series, denoted as $TS = (t(s1), t(s2), ..., t(sn))$, have the relationship of $t(s_i) = t(s_{i-1}) + \tau$, where $\tau$is the sampling interval.*

In this work, we consider the change information of a time series, so the change point of a time series is defined as follow:

DEFINITION 2 (CHANGE POINT). *A time series, denoted as $S = (s_1, s_2, ..., s_m)$, a change point is a time stamp $t_s(i)$that there is a change before and after this time stamp. Change contains the following types: mean change, variance change, frequency change, and the combination between them.*

The definition of change point set is defined as follow:

DEFINITION 3 (CHANGE POINT SET). *A time series, denoted as $S = (s_1, s_2, ..., s_m)$ The timestamps of a time series, denoted as $TS = (t(s1), t(s2), ..., t(sn))$ The change points set is denoted as $C_X = (t_x(1), t_x(2), ..., t_x(p))$ Where $t_x(i)$ denotes the change points of time series S.*

2.2 Change based Time Series Correlation

After define the time series, we define the correlation of this work as follow:

DEFINITION 4 (CHANGE BASED CORRELATION). *Suppose we have two time series: $X_1 = (x_1, x_2, ..., x_m), Y_1 = (y_1, y_2, ..., y_m)$, The change point set of X and Y are denoted as: $C_X = (t_x(1), t_x(2), ..., t_x(p))$ $C_Y = (t_y(1), t_y(2), ..., t_y(q))$ where q and p are numbers of change points for time series X and Y. Then if X and Y are correlated if and only if:*

$$\begin{cases} p = q \\ L1(C_X, C_Y) < \xi \end{cases} \tag{1}$$

where $L1(C_X, C_Y)$ denotes the L1 distance, and ξ is the threshold of Time series correlation.

3. THE APPROACH

In this section, we first propose a framework to analyze the correlation of heterogeneous time series, and then we introduce how to use hashing method to do fast searching.

3.1 Change Based Correlation Coefficient

The Change Based Correlation Coefficient can be calculated following the framework in Fig. 3. Given a time series, first, we extract the change information of the time series. In this work, we regard the change information as bit-stream. In the bit-stream, 1 denotes there is a change in this sub-series, and 0 if not. We will introduce the change based information in details in the following section.

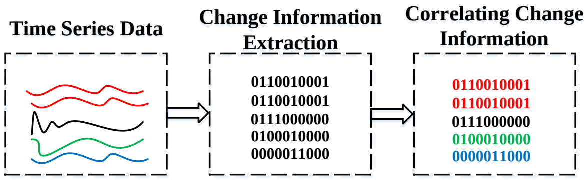

Figure 3: Overview of the Framework

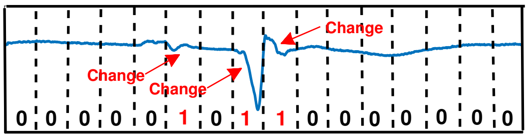

Figure 4: Change Information Extraction

After obtaining the change information of the time series, we then calculate the Jaccard similarity[8] coefficient between each other. As we defined before (Section 2), if two time series often change at the same time, they may have correlation with each other. So, here, how often denotes the value of the correlation. In other words, how many 1 do two time series both have. This is directly the Jaccard Distance. So, the Jaccard similarity between each bit-stream will be the correlation coefficient between these two time series.

3.2 Change Information Extraction

As we introduced in Section.2, change based correlation corresponds to the change information of the time series. change information of a time series is a time period information, not a time point information. As a result, in order to extract the change information of the time series, we need to find the information in small time period of the time series (A sub-series). The idea of extracting the change information is showed in Fig.4.

Given a time series $S = (s_1, s_2, ..., s_m)$, where m is the number of points in the time series. Given a subs-series length k. The change information of the time series S can be represented as a bit-stream:

$B_S = \{b_0, b_1, ..., b_n\}$, where each b_i corresponds to a sub-series of length k for the original time series S as showed in

Given a sub-series $l^j = \{s_i, s_{i+1}, s_{i+2}, ..., s_{i+k-1}\}$, where w is the length of the sub-series. Then the change information of sub-series l is denoted as follow:

$$b^l = \begin{cases} 1 & \textit{Have change in } l \\ 0 & \textit{No change in } l \end{cases} \tag{2}$$

As showed in Equ.2, the change information is the information that whether there's a change in the sub-series. In order to denote whether there's a change in the sub-series, we need to know to to detect change in the sub-series. Fig.4 shows how to extract the change information of the time series.

3.3 Change Detection

So,the problem here is: Given a sub-series
$l^j = \{s_i, s_{i+1}, s_{i+2}, ..., s_{i+k-1}\}$,
how to denote whether there is a change or not in this time

series. There are a so many time series change detection methods [15, 3] proposed in the literature.

In this work, the change detection task here is not find the change points of the time series, instead we only need to denote whether there's a change in the time series. It is pointed out that all the change point detection methods can be used here to detect change information. In our experiment, we use the the following method to detect change:

We equally divide the time series into two series:
$l^j_{Front} = \{s_i, s_{i+1}, s_{i+2}, ..., s_{i+(k-1)/2-1}\}$
and,
$l^j_{Front} = \{s_{i+(k-1)/2-1}, s_{i+(k-1)/2}, ..., s_{i+k-1}\}$.

So, regard l^j_{Front} and l^j_{Rear} as two data sampled from two distributions P_1 and P_2. So, if P_1 and P_2 are statistically the same, then we can say there's not change between each other. Otherwise, there is a change in this dataset.

Then, the problem here becomes a *Two Sample Problem* [6]. We use the Two Sample t-test [19] method to solve this problem:

Here, the t_{score} between l^j_{Front} and l^j_{Rear} can be calculated as:

$$t_{score} = \frac{\overline{l^j_{Front}} - \overline{l^j_{Rear}}}{\sigma_p \sqrt{2/k}} \tag{3}$$

where, $\overline{l^j_{Front}}$ and $\overline{l^j_{Front}}$ are the mean values of l^j_{Front} and l^j_{Front}. And σ_p is as follow:

$$\sigma_p = \frac{(k-1)\sigma^2_{l^j_{Front}} + (k-1)\sigma^2_{l^j_{Rear}}}{k-1} \tag{4}$$

Then, if $t_{score} > \alpha$, we can say that these two samples are from different distributions, and thus there is a change in the sub-series l^j.

3.4 Jaccard Similarity Coefficient

After obtaining the change information (Bit-stream) of each data, we then use Jacord Similarity Coefficient to calculate the Change Correlation of each time series.

The Jaccard Similarity [8] is defined as follow: Given two Bit-stream X and Y, the Jaccord distance is showed as follow:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{5}$$

where, $|A \cap B|$ denotes the number of bit that X and Y both 1. And $|A \cup B|$ denotes the number of bit that at least X or Y is 1. For example, given two bit stream: $X = 100111$, and $Y = 001110$. Then $|A \cap B| = 2$, and $|A \cap B| = 5$, so $J(A,B) = \frac{2}{5} = 0.4$.

3.5 Speed up Top-k Search using LSH

In many the real world problems, the scale of time series data often huge [26]. Mining such huge number of time series is a big challenge for us.

In this work, by taking the advantage of the change based coefficient, we propose to use Locality Sensitive Hashing to do large scale search in time series data.

The change information of a time series is a bit-stream (Section 3.3). We regard it as the hashing code of the original time series.

As ar result, we can directly use LSH (Locality Sensitive Hashing) search algorithm to speed up the searching process. [11]

3.6 Discuss about Sub-series Length w

In this research, the sub-series length w is a very important parameter. If the sub-series length w is too short, then the change information can not be captured. On the other hand, if the sub-series length w is too long, then there will be too much noise information.

In some cases, the value of m can be selected based on domain knowledge and experiments. In the experiment of this work, all the sub-series length are selected based on the domain knowledge.

However, in most real world situations, there are millions of time series and events, and we do not have enough domain knowledge to pre-select the values of all sub-series lengths.

In our previous research of Correlating Event with time series [16], we can auto-select the sub-series length for a time series based on the autocorrelation function [7] of the time series. Given a time series $S = (s_1, s_2, ..., s_n)$, the autocorrelation is showed as follow:

$$R(l) = E(s_i * s_{i-l}). \tag{6}$$

where l denotes the lag of the correlation. The autocorrelation function of a time series can be used to represent the energy of signals in the time series with a period of l [7]. Therefore, our length m can be assigned as the value of the first peak to include the significant signal of the time series. For more detail of this selection method, please refer [16].

4. EMPIRICAL EVALUATION

In this section, we make an empirical evaluation of our algorithm by performing a set of experiments on the synthetic data set, and several real world data sets.

4.1 Comparison Methods

In order to evaluate the effectiveness of our algorithm, we choose three time series similarity algorithms and four correlation coefficient in our experiment.

For the three similarity algorithm, we choose L1-Distance, L2-Distance [8], and DTW-Distance [21]. And for the three similarity algorithm, we choose Pearson correlation [5], which is the widely used methods for correlation mining in time series. And also two ranking based correlation: The Kendall rank correlation [14], and Spearman's rank correlation [23].

In the rest of this subsection, we briefly introduce the three similarity measures and the three Correlation measures.

4.1.1 Similarity Measures

Given a two time series $X = (x_1, x_2, ..., x_m), Y = (y_1, y_2, ..., y_m)$. The L1-distance is denoted as follow:

$$L1(X,Y) = \sum_{1}^{m} |x_i - y_i| \tag{7}$$

The L2-distance is denoted as follow:

$$L1(X,Y) = \sqrt{\sum_{1}^{m} |x_i - y_i|^2} \tag{8}$$

The third similarity measure we used is DTW distance [21], which is a famous time series similarity measure.

In order to introduce the DTW distance, we firstly construct an $m-by-m$ matrix W, where the $(i-th, j-th)$ element of the matrix W. The DTW distance is to find a path through the matrix that minimizes the total cumulative distance between X and Y. So, the optimal path is the path that minimize the warping cost:

$$DTW(X,Y) = \min \sqrt{\sum_{k=1}^{K} w_k} \tag{9}$$

where, w_k belongs to the $k-th$ element of a warping path P, which is a contiguous set of elements that represent a mapping between X and Y.

4.1.2 Correlation Measures

In this subsection, we introduce three widely used correlation measures between time series: Pearson Correlation [22], Kendall rank correlation [14], and Spearman's rank correlation [23].

The Pearson correlation method is one of the most widely used method for measuring the correlation between two time series. The Pearson correlation coefficient, denoted as ρ. can be calculated as follow:

$$\rho_{X,Y}^{Pearson} = \frac{cov(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y}$$

where cov is the covariance, σ_X is the standard deviation of X, μ_X is the mean of X and $E[*]$ denotes the expectation.

The Kendall rank correlation [14] is defined as follow:

$$\rho_{X,Y}^{Kendall} = \frac{N_c - N_d}{m(m-1)/2}$$

Where N_c is the number of concordant pairs, and N_d is the number of discordant pairs, and m is the dimension of the time series. For any pair (x_i, y_i) and x_j, y_j, where $i \neq j$, are said to be concordant both $x_i > x_j$ and $y_i > y_j$, or $x_i < x_j$ and $y_i < y_j$. Otherwise, they are discordant.

The Spearman's rank correlation [23] is defined as follow:

$$\rho_{X,Y}^{Spearman} = 1 - \frac{6\sum d_i^2}{n(n^2-1)}$$

where d_i is defined as the difference between the ranks of x_i and y_i.

4.2 Effectiveness Study on Synthetic Dataset

In this section, we introduce the experiment on the synthetic Dataset.

4.2.1 The Synthetic Dataset

Synthetic data set is very useful for evaluating algorithms and functions for data mining and machine learning models[8]. In this section, we introduce the Synthetic Dataset used in our experiment.

In this synthetic Dataset, we randomly generate 10000 time series with two patterns of time series: (1) Periodical Pattern, (2) Linear Pattern. Each time series was added with white noise [8]. Then, we separate the 10000 time series into five groups. And within each group, we add change randomly at the same time. (In each group, the change points are same.) Seven different types of changes are added randomly into each time series, the seven change types are showed in Table.1.

Table 1: **Change Types in the Synthetic Data**

Change Type
Mean Change
Variance Change
Frequency Change + Variance Change
Mean Change + Frequency Change
Frequency Change + Variance Change
Mean Change + Frequency Change + Variance Change

Table 3: Summery of Synthetic Data

DataSet	Data Size	Time Series Length
Synthetic-T0	1000	800
Synthetic-T1	1000	5000
Synthetic-T2	10000	800
Synthetic-T3	10000	5000

In order to test both the efficiency and effectiveness of change based correlation coefficient. We choose four sub-set from such dataset. We make the data set size from small to large, and also the time series length from short to long. The four sub dataset is showed in Table.3.

4.2.2 Clustering Task

In order to evaluate the performance of our correlation coefficient, we design a clustering task on the Synthetic Time series data. In this experiment, we only use the Hierarchical Clustering [8] to evaluate the performance. Because Hierarchical Clustering is very sensitive to the distance measure used, it is good for evaluating the distance measure.

Two evaluation methods are used for testing the clustering result: Accuracy [8], which is calculated as the percentage of target objects clustered into the correct clusters; and Normalized Mutual Information (NMI) [8], which is one of the most popular evaluation methods to evaluate the quality of clustering results.

From Table.2, we can see that, for the change correlation coefficient, the clustering result performance is better for the high dimensional data set. This is because that for high dimensional time series data, the proposed coefficient can extract more change information from the time series data and also make more accurate evaluation of the correlation. From this point of view, change based correlation is more suitable for the high dimensional time series data set. On the other hand, Change based correlation can obtain more accuracy results in different dataset compared with both the similarity method and the correlation coefficient. So, the result of clustering show the effectiveness of our coefficient.

Fig.5 shows the Execution time of the clustering task on each data set. From the result, we can see that change based correlation performed much faster than other algorithms. This is because, we only calculate the correlation on the extracted change information (Bit-stream), the calculating of change based correlation measure can be much faster than other similarity and correlation methods.

4.2.3 Top-K Searching Task

Table 2: Clustering Performance on Synthetic Data Set

Dataset	Measure	Proposed	$L1$	$L2$	DTW	Pearson	Kendall	Spearman
Sythetic-T0	Accuracy	**.854 ± .032**	.241 ± .098	.281 ± .012	.230 ± .061	.309 ± .140	.353 ± .026	.297 ± .036
	NMI	**.808 ± .034**	.026 ± .067	.076 ± .023	.028 ± .075	.140 ± .55	.395 ± .015	.150 ± .088
Sythetic-T1	Accuracy	**.838 ± .025**	.247 ± .026	.262 ± .032	.283 ± .012	.240 ± .018	.374 ± .067	.341 ± .067
	NMI	**.701 ± .030**	.003 ± .062	.057 ± .043	.064 ± .036	.046 ± .084	.404 ± .023	.230 ± .042
Sythetic-T2	Accuracy	**.806 ± .029**	.254 ± .066	.263 ± .080	.304 ± .022	.388 ± .024	.384 ± .032	.502 ± .182
	NMI	**.889 ± .012**	.028 ± .042	.056 ± .056	.054 ± .032	.303 ± .064	.394 ± .052	.450 ± .049
Sythetic-T3	Accuracy	**.856 ± .077**	.225 ± .028	.229 ± .034	.284 ± .062	.454 ± .032	.454 ± .032	.454 ± .032
	NMI	**.891 ± .017**	.021 ± .040	.041 ± .043	.086 ± .038	.454 ± .032	.454 ± .032	.454 ± .032

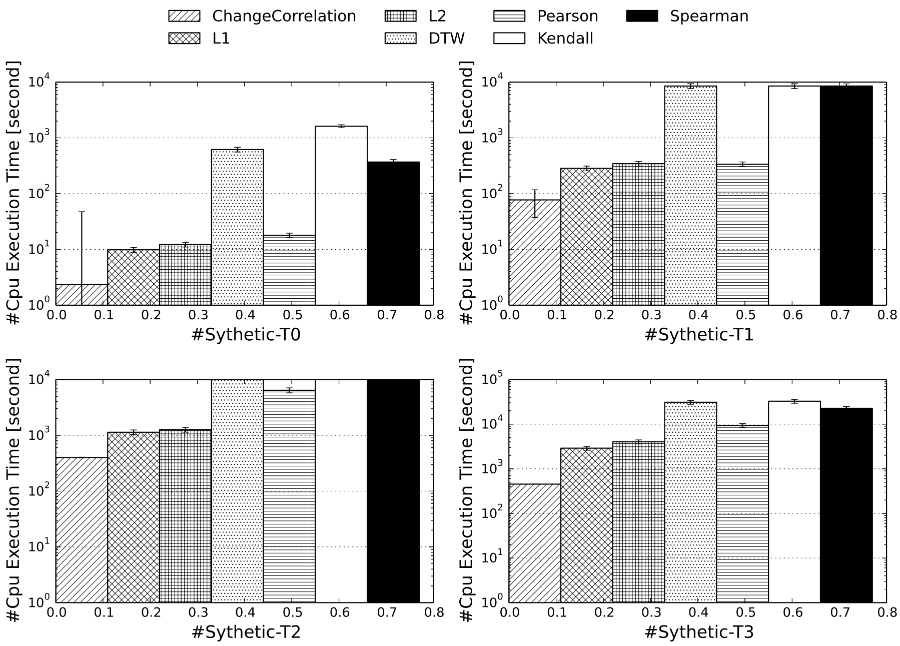

Figure 5: Top-k Nearest Neighbor Search

We compute precision and recall [24] on the data-set using the LSH-based method. And for other distance measure, we use the naive Top-K searching method.

For precision and recall, if the Searched time series is in the same cluster of the query time series, we regard it as a relevant items, and vice verse. We range the K from 1 to the cluster size.

For each top-k Nearest Neighbors search, the precision can be calculated as follow:

$$Precision = \frac{relevant\ item}{K} \tag{10}$$

and the recall can be calculated as follow:

$$Precision = \frac{relevant\ item}{Relevant\ Cluster\ Size} \tag{11}$$

The plots for all the three datasets are shown in Figure.6. We can clearly see that our proposed Change-based Correlation method gives significantly higher precision recall curves than other similarity and correlation methods. In addition the results are consistent across datasets.

Fig.7 shows the execution time by vary the data size and the time series length. In left one of Fig.7, we fix the value of time series length, and vary the data size n. We can see that the CPU execution time of other similarity methods increased sharply by enlarging the data size. And, the change based correlation do not change so much by enlarge the data size.

In right of Fig. 7, we fix the size of data size, and vary the value of time series length. Based on the results, we can see that the running time of the proposed change based method with LSH doesn't so much with the increase of time series length, while other methods increase by enlarging the time series length.

4.3 Effectiveness Study on Real Datasets

In this section, we will compare the proposed algorithm with the baseline algorithms on two real data sets.

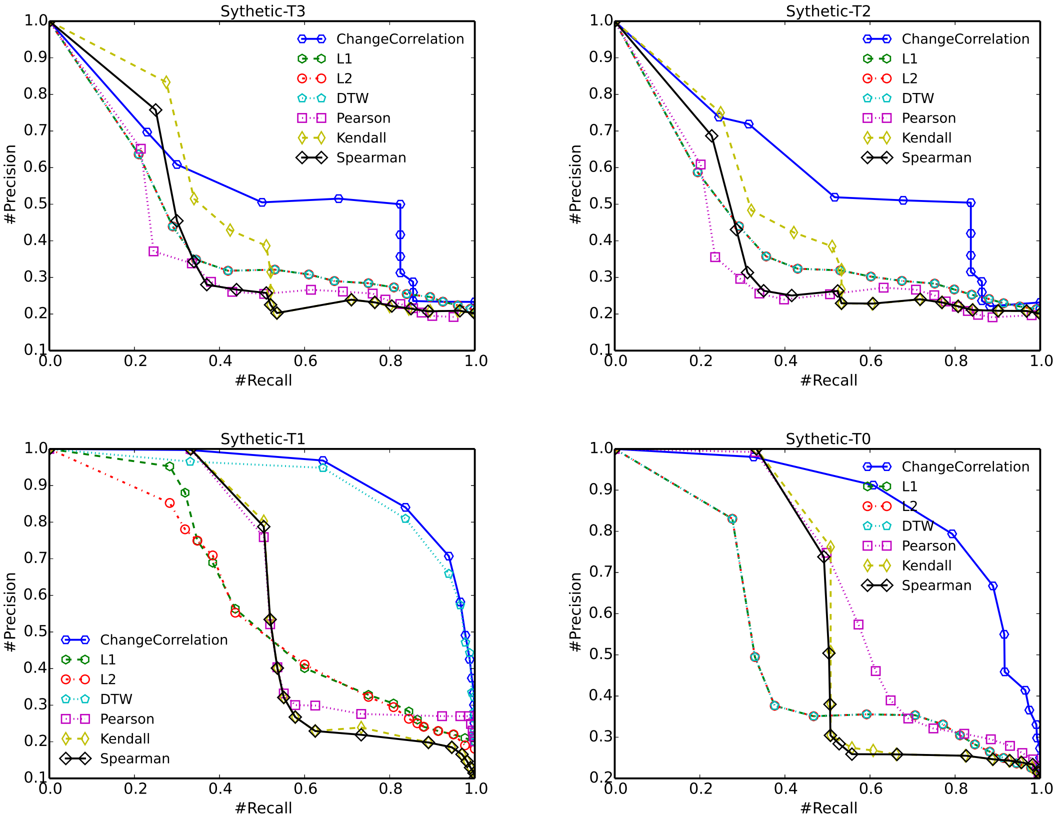

Figure 6: Precision Recall Curve for Different Algorithms

4.3.1 *Electrocardiogram Data set*

The first real world dataset is ECG (Electrocardiogram) time series data set. This data set comes from the the UCR time series Data set Archive [4]. We choose four ECG data set there as showed in Table. 4. And the ground truth comes from the UCR data set itself.

For the clustering task, we use the Hierarchical Clustering [8] to evaluate the performance as before. From Table.5, we can see that, for the change based correlation can obtain more accuracy results in these four ECG data set compared with both the similarity method and the correlation coefficient. So, the result of clustering show the effectiveness of our coefficient.

For the Top-K searching task. The plots for all the four ECG dataset datasets are shown in Figure.8. We can clearly see that our proposed Change-based Correlation method gives significantly higher precision recall curves than other similarity and correlation methods. In addition the results are consistent across datasets. This demonstrate the effectiveness of change-based correlation coefficient and the corresponding LSH search algorithm.

4.3.2 *HPC Thread Time Series Data set*

The second real world dataset is from the HPC-tool Kit Dataset. This data set provided by HPCToolkit by analysis High performance computer during running. There are two types of thread in these two data set: main thread and other thread. So, the experiment here is based on the ground truth of two different class of thread.

Table 6: Summary of the HPC Time Series Data Set

Data Set	Data Size	Time Series Length
Single PC	24	4096
MADNESS	264	32768

We also use the Hierarchical Clustering [8] to evaluate the performance as before. From Table.7, we can see that, for the change based correlation can obtain more accuracy results in the two HPC Thread Time series data set compared with both the similarity method and the correlation coefficient. While we can see that DTW method can also get high accuracy, this because in data set, most of the thread in same class often in same state. However, as we said before, in some cases, thread in same class, the state can be different.

The precision recall curve for the HPC Thread TIme Series data is showed in Figure 9. We can clearly see that our proposed Change-based Correlation method gives higher precision recall curves than other similarity and correlation methods. Also, DTW method can get good result too. In addition the results are consistent across datasets. This demonstrate the effectiveness of change-based correlation coefficient and the corresponding LSH search algorithm.

5. RELATED WORK

In this section, we brief introduce some related works of our research.

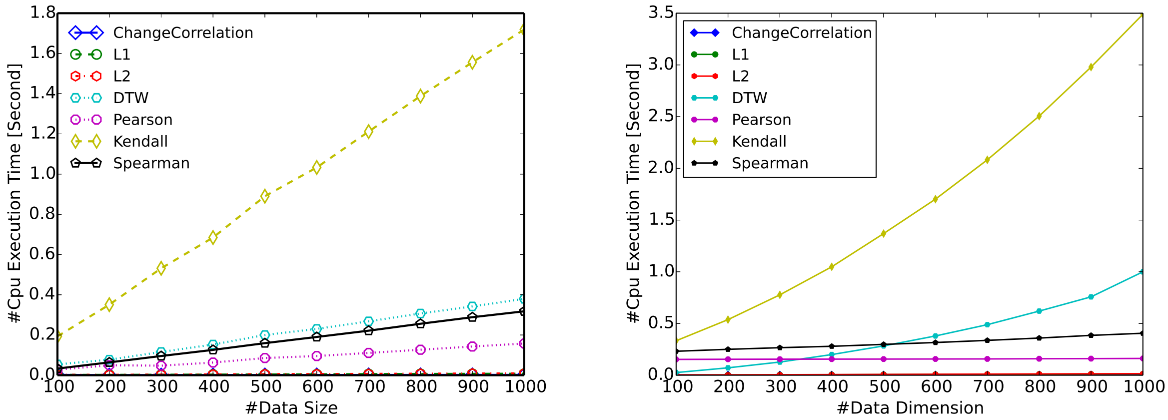

Figure 7: Efficiency by varying data size and dimension size

Table 4: Summary of the Four ECG Data Set

Data Set	Data Size	Time Series Length	Class Numver
CinC_ECG_torso	1380	1639	4
ECGFiveDays	861	136	2
TwoLeadECG	1139	82	2
ECG5000	4500	140	5

5.1 Correlation between Time Series Data

Correlation between two time series has been widely studied, and some of them have been included in text books [12]. Pearson Correlation [5] is a basic correlation measure between time series, which has been widely used in practice [35]. Some extensions of Pearson correlation are also widely used. For example, lagged correlation is an extension to correlate a lagged dataset with another unlagged dataset using the Pearson product-moment method. In [32], the author uses the lagged-correlation to estimate the lead relationship between a set of time series. Because Pearson correlation is sensitive outliers in data set, Spearman Rank correlation and Kendall Rank correlation have been used in some scenarios [27] to overcome the drawbacks of Pearson correlation. In Spearman correlation, data is first sorted and each value assigned a rank, e.g., 1 is assigned to the lowest value. Spearman Rank correlation is calculated by taking the Pearson product-moment correlation of the ranks of the datasets. Kendall correlation is used to measure the similarity of the orderings of the data when ranked by each of data values. Because there is no ordering relationship among the different events, the above rank based algorithms cannot be directly used in our scenario.

5.2 Change Point Detection

The problem of change detection has been studied for a long time, and various methods such as CUSUM (cumulated summation) [2], wavelet analysis [13], inflection point search [9], and Gaussian mixtures [33] have been proposed. These algorithms can be used in our method. However, our provided method can quickly detect the boolean problem of whether there is a change in the time period. We do not need to find the time series change point very accurate.

6. CONCLUSION AND FUTURE WORKS

Calculating the correlation between different time series is an important data mining task. By using correlation method, we can find hidden relationships between different time series. However, in most real world problems, time series often have very different patterns (e.g. Periodical, Linear, random, etc.). And there also correlations between such heterogeneous time series, and the correlation between these time series is also a very import problem. Despite their importance, there has been little previous work addressing the correlation between two types of heterogeneous time series data.

In this paper, we propose an approach that is capable of (1) evaluating evaluate the correlation between heterogeneous time series (time series with different patterns), and (2) dealing with very large top-k searching problems. The experimental results on Synthetic data sets and real world data set show the effectiveness and efficiency of our algorithms.

7. REFERENCES

[1] L. Adhianto, S. Banerjee, M. Fagan, M. Krentel, G. Marin, J. Mellor-Crummey, and N. R. Tallent. Hpctoolkit: Tools for performance analysis of optimized parallel programs. *Concurrency and Computation: Practice and Experience*, 22(6):685–701, 2010.

[2] M. Basseville, I. V. Nikiforov, et al. *Detection of abrupt changes: theory and application*, volume 104. Prentice Hall Englewood Cliffs, 1993.

[3] X. C. Chen, K. Steinhaeuser, S. Boriah, S. Chatterjee, and V. Kumar. Contextual time series change detection. In *SDM*, pages 503–511. SIAM, 2013.

Table 5: Clustering Performance on Synthetic ECG Data Set From UCR Time Series Archive

Dataset	Measure	Proposed	$L1$	$L2$	DTW	Pearson	Kendall	Spearman
CinC_ECG_torso	Accuracy	**.839 ± .011**	.667 ± .068	.557 ± .012	.610 ± .061	.531 ± .140	.507 ± .019	.504 ± .013
	NMI	**.489 ± .019**	.236 ± .035	.019 ± .023	.010 ± .075	.280 ± .55	.150 ± .015	.049 ± .012
EGG_5000	Accuracy	**.538 ± .025**	.247 ± .026	.262 ± .032	.283 ± .012	.240 ± .018	.374 ± .067	.341 ± .067
	NMI	**.401 ± .030**	.003 ± .062	.057 ± .043	.064 ± .036	.046 ± .084	.404 ± .023	.230 ± .042
TwoLeadECG	Accuracy	**.810 ± .029**	.504 ± .066	.538 ± .080	.620 ± .022	.528 ± .064	.531 ± .052	.519 ± .049
	NMI	**.680 ± .012**	.081 ± .042	.043 ± .056	.137 ± .032	.047 ± .064	.062 ± .052	.074 ± .049
ECGFiveDays	Accuracy	**.832 ± .077**	.502 ± .028	.527 ± .034	.615 ± .062	.506 ± .032	.547 ± .032	.519 ± .032
	NMI	**.765 ± .017**	.002 ± .040	.002 ± .043	.361 ± .038	.075 ± .032	.023 ± .032	.086 ± .032

Table 7: Clustering Performance on Synthetic ECG Data Set From UCR Time Series Archive

Dataset	Measure	Proposed	$L1$	$L2$	DTW	Pearson	Kendall	Spearman
Single PC	Accuracy	**.917 ± .011**	.835 ± .068	.815 ± .012	.870 ± .061	.501 ± .140	.527 ± .019	.514 ± .013
	NMI	**.889 ± .019**	.736 ± .035	.719 ± .023	.860 ± .075	.380 ± .55	.350 ± .015	.349 ± .012
MADNESS	Accuracy	**.938 ± .025**	.927 ± .026	.922 ± .032	.935 ± .012	.457 ± .018	.474 ± .067	.541 ± .067
	NMI	**.861 ± .030**	.853 ± .062	.857 ± .043	.864 ± .036	.346 ± .084	.404 ± .423	.230 ± .442

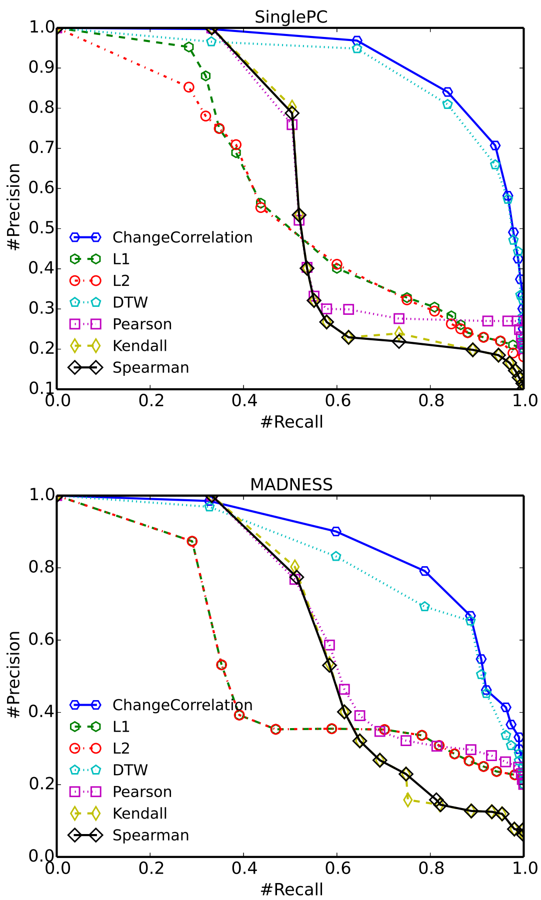

Figure 9: Precision Recall Curve (higher is better). We compare Change based correlation coefficient with other methods.

[4] Y. Chen, E. Keogh, B. Hu, N. Begum, A. Bagnall, A. Mueen, and G. Batista. The ucr time series classification archive, July 2015. `www.cs.ucr.edu/~eamonn/time_series_data/`.

[5] J. Cohen. Statistical power analysis for the behavioral sciences. 1988.

[6] A. Gretton, K. M. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola. A kernel method for the two-sample-problem. In *Advances in neural information processing systems*, pages 513–520, 2006.

[7] J. D. Hamilton. *Time series analysis*, volume 2. Princeton university press Princeton, 1994.

[8] J. Han, M. Kamber, and J. Pei. *Data mining: concepts and techniques: concepts and techniques*. Elsevier, 2011.

[9] S. Hirano and S. Tsumoto. Mining similar temporal patterns in long time-series data and its application to medicine. In *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*, pages 219–226. IEEE, 2002.

[10] N. J. Holter. New method for heart studies continuous electrocardiography of active subjects over long periods is now practical. *Science*, 134(3486):1214–1220, 1961.

[11] P. Indyk and R. Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 604–613. ACM, 1998.

[12] R. A. Johnson and D. W. Wichern. *Applied multivariate statistical analysis*. Pearson, 2007.

[13] S. Kadambe and G. F. Boudreaux-Bartels.

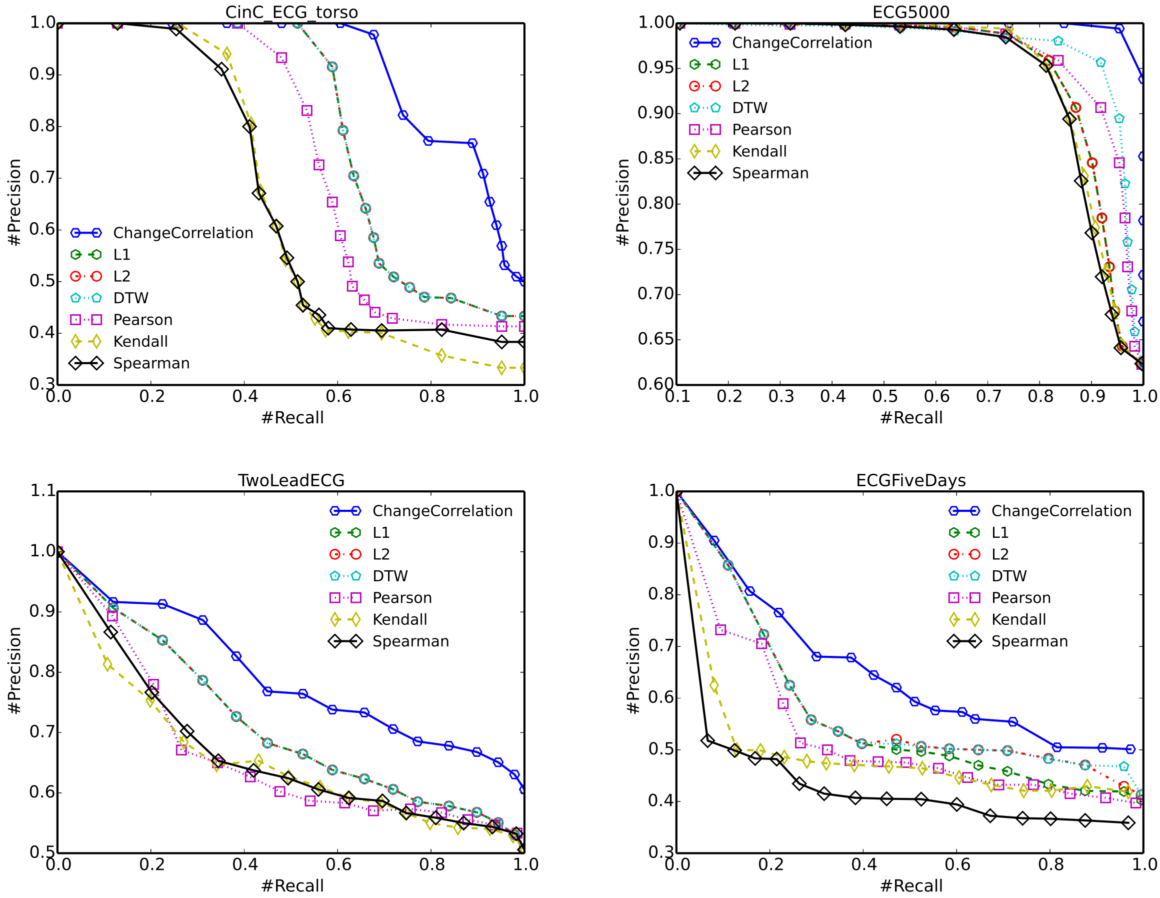

Figure 8: Precision Recall Curve (higher is better). We compare Change based correlation coefficient with other methods.

Application of the wavelet transform for pitch detection of speech signals. *IEEE Transactions on Information Theory*, 38(2):917–924, 1992.

[14] M. G. Kendall. A new measure of rank correlation. *Biometrika*, pages 81–93, 1938.

[15] S. Liu, M. Yamada, N. Collier, and M. Sugiyama. Change-point detection in time-series data by relative density-ratio estimation. *Neural Networks*, 43:72–83, 2013.

[16] C. Luo, J.-G. Lou, Q. Lin, Q. Fu, R. Ding, D. Zhang, and Z. Wang. Correlating events with time series for incident diagnosis. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1583–1592. ACM, 2014.

[17] H. J. L. Marriott and G. S. Wagner. *Practical electrocardiography*. Williams & Wilkins Baltimore, 1988.

[18] C. McCurdy and J. Vetter. Memphis: Finding and fixing numa-related performance problems on multi-core platforms. In *Performance Analysis of Systems & Software (ISPASS), 2010 IEEE International Symposium on*, pages 87–96. IEEE, 2010.

[19] D. S. Moore. *The basic practice of statistics*, volume 2. WH Freeman New York, 2007.

[20] D. W. Mount and D. W. Mount. *Bioinformatics: sequence and genome analysis*, volume 2. Cold spring harbor laboratory press New York:, 2001.

[21] M. Müller. Dynamic time warping. *Information retrieval for music and motion*, pages 69–84, 2007.

[22] K. Pearson. *Mathematical contributions to the theory of evolution*, volume 13. Dulau and co., 1904.

[23] W. Pirie. Spearman rank correlation coefficient. *Encyclopedia of statistical sciences*, 1988.

[24] D. M. Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness & correlation. *JMLT*, 2(1):37–63, 2011.

[25] L. Rabiner and B.-H. Juang. Fundamentals of speech recognition. 1993.

[26] T. Rakthanmanon, B. Campana, A. Mueen, G. Batista, B. Westover, Q. Zhu, J. Zakaria, and E. Keogh. Searching and mining trillions of time series subsequences under dynamic time warping. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 262–270. ACM, 2012.

[27] B. Rosner. *Fundamentals of biostatistics*. Cengage Learning, 2010.

[28] M. Sonka, V. Hlavac, and R. Boyle. *Image processing, analysis, and machine vision*. Cengage Learning, 2014.

[29] C. Sun, H. Zhang, J.-G. Lou, H. Zhang, Q. Wang,

D. Zhang, and S.-C. Khoo. Querying sequential software engineering data. In *Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering*, pages 700–710. ACM, 2014.

[30] N. R. Tallent and J. M. Mellor-Crummey. Effective performance measurement and analysis of multithreaded applications. In *ACM Sigplan Notices*, volume 44, pages 229–240. ACM, 2009.

[31] L. P. Tilley et al. *Essentials of canine and feline electrocardiography.* CV Mosby., 1979.

[32] D. Wu, Y. Ke, J. X. Yu, S. Y. Philip, and L. Chen. Detecting leaders from correlated time series. In *Database Systems for Advanced Applications*, pages 352–367. Springer, 2010.

[33] K. Yamanishi and J.-i. Takeuchi. A unifying framework for detecting outliers and change points from non-stationary time series data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 676–681. ACM, 2002.

[34] M.-H. Yang, D. J. Kriegman, and N. Ahuja. Detecting faces in images: A survey. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(1):34–58, 2002.

[35] Y. Zhu and D. Shasha. Statstream: Statistical monitoring of thousands of data streams in real time. In *VLDB*, pages 358–369. VLDB Endowment, 2002.