Characteristic-Based Clustering for Time Series Data

Article *in* Data Mining and Knowledge Discovery · September 2006

DOI: 10.1007/s10618-005-0039-x · Source: DBLP

Xiaozhe Wang

19 PUBLICATIONS **489** CITATIONS

SEE PROFILE

Kate Smith-Miles
University of Melbourne

295 PUBLICATIONS **6,113** CITATIONS

SEE PROFILE

Rob J Hyndman
Monash University (Australia)

278 PUBLICATIONS **15,747** CITATIONS

SEE PROFILE

Data Mining and Knowledge Discovery, 13, 335–364, 2006
© 2006 Springer Science + Business Media, LLC. Manufactured in the United States.
DOI: 10.1007/s10618-005-0039-x

Characteristic-Based Clustering for Time Series Data

XIAOZHE WANG* catherine.wang@infotech.monash.edu.au
KATE SMITH kate.smith@infotech.monash.edu.au
Faculty of Information Technology, Monash University, Clayton, Victoria 3800, Australia

ROB HYNDMAN rob.hyndman@buseco.monash.edu.au
Department of Econometrics and Business Statistics, Monash University, Clayton, Victoria 3800, Australia

Received July 12, 2005; Accepted December 12, 2005

Published online: 16 May 2006

Abstract. With the growing importance of time series clustering research, particularly for similarity searches amongst long time series such as those arising in medicine or finance, it is critical for us to find a way to resolve the outstanding problems that make most clustering methods impractical under certain circumstances. When the time series is very long, some clustering algorithms may fail because the very notation of similarity is dubious in high dimension space; many methods cannot handle missing data when the clustering is based on a distance metric.

This paper proposes a method for clustering of time series based on their *structural* characteristics. Unlike other alternatives, this method does not cluster point values using a distance metric, rather it clusters based on global features extracted from the time series. The feature measures are obtained from each individual series and can be fed into arbitrary clustering algorithms, including an unsupervised neural network algorithm, self-organizing map, or hierarchal clustering algorithm.

Global measures describing the time series are obtained by applying statistical operations that best capture the underlying characteristics: trend, seasonality, periodicity, serial correlation, skewness, kurtosis, chaos, nonlinearity, and self-similarity. Since the method clusters using extracted global measures, it reduces the dimensionality of the time series and is much less sensitive to missing or noisy data. We further provide a search mechanism to find the best selection from the feature set that should be used as the clustering inputs.

The proposed technique has been tested using benchmark time series datasets previously reported for time series clustering and a set of time series datasets with known characteristics. The empirical results show that our approach is able to yield meaningful clusters. The resulting clusters are similar to those produced by other methods, but with some promising and interesting variations that can be intuitively explained with knowledge of the global characteristics of the time series.

Keywords: time series clustering, clustering, global characteristics, feature measures, dimensionality reduction

1. Introduction

The mining of time series data has attracted great attention in the data mining community in recent years (Bradley and Fayyad, 1998; Halkidi et al., 2001; Kalpakis et al., 2001; Keogh et al., 2003; Wang and Wang, 2000). Each time series, while consisting of a large number of data points, can also be seen as a single object. Classification and clustering of such complex "objects" may be particularly beneficial for the areas of process control,

*Corresponding author.

intrusion detection, and character recognition (Faloutsos et al., 1994). Clustering time series and other sequences of data has become an important topic, motivated by several research challenges including similarity search of medical and astronomical sequences (Scargle, 2000), as well as the challenge of developing methods to recognize dynamic changes in time series.

Most clustering methods for time series data typically require many conditions to be useful, and unfortunately, those methods may become impractical under certain circumstances. This paper proposes a characteristic-based method for clustering of time series. Unlike other alternatives, this method does not cluster point values using a distance measure, but it clusters global features extracted from each time series. The feature measures are obtained from each individual series and then fed into the clustering algorithm of choice, including an unsupervised neural network algorithm, Self Organizing Map (SOM), for visualization and interpretation.

Below we list some of the advantages of our approach, which will be explained in more detail along with empirical results in later sections.

- When the time series is very long (high dimensionality), some clustering algorithms become intractable. By applying dimensionality reduction via feature extraction, we are able to cluster long length time series very efficiently;
- When the clustering algorithm is based on a distance metric (i.e. Euclidean distance), it cannot handle time series with missing data or of different lengths if actual points are used as inputs. However, by extracting a set of measures from the original time series we simply bypass this problem.
- When clustering long time series datasets, the majority of existing time series clustering methods, which mostly concentrate on shape-based datasets, may produce a low quality clustering results. By clustering based on global characteristics measures extracted from time series, our approach can yield better clusters in the structure-level clustering context.

Because the best feature set may differ from domain to domain, our proposed approach incorporates a built-in search mechanism. Our greedy search method can be applied to discover global measures that lead to the best clustering result. As a result, our approach can find more intuitive clusters.

The remainder of the paper is organized as follows. Section 2 presents the background of time series clustering and the motivation for our Characteristic-Based Clustering (CBC) approach. Section 3 describes various feature measures used to extract the global characteristics of each time series. Section 4 describes the clustering algorithms and presents clustering results tested on some benchmark time series datasets to demonstrate the suitability of our proposed method. An unsupervised learning algorithm, SOM, is also illustrated in terms of its utility in visualization. Section 5 presents the search mechanism and experimental evaluation on both synthetic and real-world datasets. Finally, Section 6 concludes the paper and outlines some directions for future research.

2. Background and motivation

In this section, we will first review the background work on time series clustering and feature extraction that inspired our research. Then, our motivation for research on time series clustering based on data characteristics is more fully developed.

2.1. *Time series clustering*

There are two main categories in time series clustering summarized by Keogh et al. (2003). "Whole clustering" is the clustering performed on many individual time series to group similar series into clusters. "Subsequence clustering" is based on sliding window extractions of a single time series and aims to find similarity and differences among different time windows. Two levels of similarity in clustering are classified as "shape level" and "structure level". The former is focusing on short-length clustering, for example gene expression profiles or individual heartbeats; and the latter is measuring similarity based on high level structure for long-length series, for example an hour's worth of ECGs or a yearly meteorological data.

Many clustering algorithms have been applied to the raw time series data and different measures have been used for measuring the similarity between data points. Some of the popular methods are briefly discussed below:

- *Algorithms*: *K*-means clustering is the most commonly used clustering algorithm (Bradley and Fayyad, 1998; Halkidi et al., 2001), with the number of clusters, *k*, specified by the user. Hierarchical clustering generates a nested hierarchy of similar groups of time series according to a pairwise distance matrix of the series (Keogh et al., 2003). One advantage of hierarchical clustering is that the number of clusters is not required as a parameter. Both of these clustering approaches, however, require that the length of each time series be identical due to the Euclidean distance calculation requirement, and are unable to deal effectively with long time series due to poor scalability.
- *Distance Measures*: Euclidean distance, the most commonly used metric (Agrawal et al., 1993; Chan and Fu, 1999; Chu and Wong, 1999; Faloutsos et al., 1994; Keogh et al., 2001; Popivanov and Miller, 2002), is certainly not the only metric available for measuring the similarity between data series. Autocorrelation has been proposed (Wang and Wang, 2000), along with a variety of other measures in recent years, including cepstrum (Kalpakis et al., 2001), piecewise normalization (Indyk et al., 2000), cosine wavelets (Huntala et al., 1999), and piecewise probabilistic measures (Keogh and Smyth, 1997). However, the survey and empirical comparison in Keogh and Kasetty (2002) revealed that the Euclidean distance metric still performs favorably compared to others when tested on the same datasets. Dynamic Time Warping (DTW) has been applied in time series mining to resolve the difficulty in clustering time series of variable lengths in Euclidean space or containing possible out-of-phase similarities (Berndt and Clifford, 1994; Ratanamahatana and Keogh, 2005).

While these methods have been quite effective in clustering moderate-length time series, there are some well-recognized drawbacks of the existing approaches. Euclidean distance is the most common method for discerning similarity in time series clustering, and it requires that the time series being compared are of exactly the same dimensionality (length). Even though hierarchical clustering is one of the most widely used approaches, it is restricted to small datasets due to its quadratic computational complexity (Keogh et al., 2003). *K*-means is a faster method (Bradley and Fayyad, 1998) than hierarchical clustering, but the number of clusters has to be pre-assigned, which is impractical in obtaining natural clustering results. DTW can assist with clustering of different length time series if there is no missing data (Ratanamahatana and Keogh, 2005) and in any case it requires quadratic computation.

2.2. Feature extraction for time series data

As mentioned at the beginning of this section, for long time series, the shape-based similarity clustering typically produces poor quality in clustering results. Therefore, structure level similarity measuring for time series data based on global features or model parameters extraction has been proposed by several authors.

- Nanopoulos extracted four basic statistical features from Control Chart Pattern data and used them as input in a multi-layer perceptron neural network for time series classification (Nanopoulos et al., 2001). Their experimental results showed the robustness of the method against noise and time series length compared to other methods using each and every data point for the same purpose.
- By using two popular feature extraction techniques, the Discrete Wavelet Transform and the Discrete Fourier Transform, Mörchen has demonstrated the advantages of feature extraction for time series clustering in terms of computational efficiency, and it showed to improve clustering on a benchmark dataset (Mörchen, 2003).
- For a given time series, the parameters of the AutoRegression Moving Average (ARMA) model are estimated and used as a limited dimensional vector for the original time series in a classification problem (Deng et al., 1997). However, using ARMA parameters is not a reliable method because different sets of parameters can be obtained even from time series with similar structure that could affect the clustering results dramatically.
- As demonstrated on real datasets by Ge and Smyth (2000), an approach for time series pattern matching based on segmental semi-Markov models has shown usefulness, flexibility, and accuracy. The time series is modeled as 'k' distinct segments with constraints on how the segments are "linked" as the representation of the data before applying a viterbi-like algorithm to compute the similarity.
- Compression-based Dissimilarity Measures (CDM) is proposed by Keogh and others to compare long time series structure using co-compressibility as a dissimilarity measure (Keogh et al., 2004). This measure can then be directly used in data mining algorithms, such as hierarchical clustering. Their extensive experiments have demonstrated the ability in handling different-length and missing-value time series.

While the above works have shown utility in certain domains, most of them have high computational complexity and require certain conditions for the method to be useful in computation. For instance, in CDM, a time series needs to be converted to another representation, such as Symbolic Aggregate ApproXimation (Lin et al., 2004).

Given the limitations of the above, we seek a method that is simple, flexible, and accurate.

2.3. Motivation for the approach

An earlier part of the paper has addressed the limitations of existing clustering approaches and similarity measures, and the inspiration from time series feature extraction idea. We wish to provide a method for clustering time series of varying lengths, which is robust to missing data.

This approach is a departure from the more common method of clustering time series based on distance measures within the space determined by the actual point values of the time series. Regardless of the length of the time series and missing values, a finite set of statistical measures will be used to capture the global nature of the time series. Following the feature extraction idea, we take the path on using statistical measures as identified features. Many different methods of features extraction have been introduced by other researchers to summarize time series data for various purposes. For example, in speech signals classification, a variety of different features including statistical and specific speech data features were used as inputs to a classification model (Dellaert et al., 1996). Their experimental results revealed the advantage of classification based on speech signal features. Nanopoulos and others (Nanopoulos et al., 2001) also suggested four different statistical time series measures for first-order and second-order decompositions; thus, a total of eight features were used to form the input vector for a neural network classifier for a control chart classification application.

The features to be identified should carry summarized information of the time series, which capture the 'global picture' of the data. The set of features identified in our research are different and more expressive than other related works. By investigating a thorough literature review on time series quantitative statistical feature, we propose a novel set of feature measures that can best capture the global characteristic of the time series; both classical statistical measures and advanced special measures are also included. The features of trend, seasonality, periodic, serial correlation, skewness, and kurtosis have been widely used as exemplary measures in many time series feature-based research (Armstrong, 2001). Some advanced features are derived from the research on relatively new phenomena, which include non-linearity structure, self-similarity, and chaos. As a result, a novel set of time series characteristics features are extracted as measures.

The feature extraction process can also be considered as a dimensionality reduction procedure in time series data mining. Extracting the summarized characteristics of the time series can provide a more meaningful dimensionality reduction compared to other methods. By applying a statistical treatment to the analysis of time series data, datasets with long-length or different-length time series are pre-processed to produce a limited number of measures and are less sensitive to noise. These features concisely represent the relevant characteristics of each time series as a finite set of inputs to a clustering algorithm that can then discern similarity and differences between the time series. The outcome of feature extraction is a set of measures that can be fed into any clustering techniques of choice.

Considering the different nature of application domains and specialized features in time series data, we are not claiming that the extracted feature set is a 'universal' solution and is applicable to any problem domain. Instead, a search mechanism is embedded in our method to optimize the measures for different types of time series data. Once the optimized measures are found for one type of pattern, our approach can then be applied directly to the same application domain with greater confidence.

3. Global characteristic measures

A univariate time series is the simplest form of temporal data and is a sequence of real numbers collected regularly in time, where each number represents a value. We represent a time series as an ordered set of n real-valued variables $Y_t = Y_1, \ldots, Y_n$. Time series can be described using a variety of qualitative terms such as seasonal, trending, noisy, non-linear, chaos, etc. As mentioned in the last section of our research motivation, there are nine classical and advanced statistical features describing a time series' global characteristics. They are: trend, seasonality, periodicity, serial correlation, skewness, kurtosis, non-linearity, self-similarity, and chaos. This collection of measures is quantified descriptors and can help provide a rich portrait of the nature of a time series.

In time series analysis, decomposition is a critical step to transform the series into a format for statistical measuring (Hamilton, 1994). Therefore, to obtain a precise and comprehensive calibration, some measures are calculated on both the raw time series data Y_t (referring as 'RAW' data), as well as the remaining time series after de-trending and de-seasonalizing Y_t' (referring as "Trend and Seasonally Adjusted (TSA)" data). But some features can only be calculated on raw data to obtain meaningful measures, such as periodicity, etc. As exhibited in Table 1, a total of thirteen measures are extracted (marked with "√") from each time series including seven on the RAW data and six on the TSA data. (Detailed explanation of the choice of extracting features from RAW or TSA data is discussed later in Sections 3.1 to 3.8.) These measures later become inputs to the clustering process. The thirteen measures are a finite set used to quantify the global characteristics of any time series, regardless of its length and missing values.

For each of the features described below, we have attempted to find the most appropriate way to measure the presence of the feature, and ultimately normalize the metric to [0, 1] to indicate the degree of presence of the feature. A measure near 0 for a certain time series indicates an absence of the feature, while a measure near 1 indicates a strong presence of the feature. The calculation of the measures and scaling transformations has been coded using *R* language (see www.r-project.org).

Table 1 Summary of identified feature measures.

Feature	RAW data	TSA data
Trend		√
Seasonality		√
Serial Correlation	√	√
Non-linearity	√	√
Skewness	√	√
Kurtosis	√	√
Self-similarity	√	
Chaotic	√	
Periodicity (frequency)	√	

3.1. Trend and seasonality

Trend and seasonality are common features of time series, and it is natural to characterize a time series by its degree of trend and seasonality. In addition, once the trend and seasonality of a time series has been measured, we can de-trend and de-seasonalize the time series to enable additional features such as noise or chaos to be more easily detectable.

A trend pattern exists when there is a long-term change in the mean level (Makridakis et al., 1998). To estimate the trend, we can use a smooth nonparametric method, such as the penalized regression spline.

A seasonal pattern exists when a time series is influenced by seasonal factors, such as month of the year or day of the week. The seasonality of a time series is defined as a pattern that repeats itself over fixed intervals of time (Makridakis et al., 1998). In general, the seasonality can be found by identifying a large autocorrelation coefficient or a large partial autocorrelation coefficient at the seasonal lag.

Let Y_t be original data, X_t be de-trended data after transformation $X_t = Y_t^* - T_t$, Z_t be de-seasonalized data after transformation $Z_t = Y_t^* - S_t$, and the remainder series is defined as $Y_t' = Y_t^* - T_t - S_t$, which is the time series after trend and seasonality adjusting. As such, the trend and seasonality measures are extracted from the TSA data. Then the suitable measure of trend is $1-\frac{\mathrm{Var}(Y_t')}{\mathrm{Var}(Z_t)}$, and the measure of seasonality $1-\frac{\mathrm{Var}(Y_t')}{\mathrm{Var}(X_t)}$.

There are three main reasons for making a transformation after plotting the data: (a) to stabilize the variance, (b) to make the seasonal effect additive, and (c) to make the data normally distributed (Chatfield, 1996). The two most popularly used transformations, logarithmic and square-root, are special cases of the class of Box-Cox transformation (Box and Cox, 1964), which is used for the 'normal distribution' purpose.

Given a time series Y_t and a transformation parameter λ, the transformed series Y_t^* is $Y_t^* = (Y_t^\lambda - 1)/\lambda \quad \lambda \neq 0$. This transformation applies to situations in which the dependent variable is known to be positive. The effective power transformation when $\lambda \neq 0$ (Chatfield, 1996) is introduced in our following decomposition procedure.

We have used the basic decomposition model in Chapter 3 of Makridakis et al. (1998):

- $Y_t^* = T_t + S_t + E_t$, where Y_t^* denotes the series after Box-Cox transformation, at time t, T_t denotes the trend, S_t denotes the seasonal component, and E_t is the irregular (or remainder) component;
- For a given transformation parameter λ, if the data are seasonal, which is identified when a known parameter f (frequency or periodicity which is discussed in Section 3.2) from input data satisfies as $f > 1$, the decomposition is carried out using the STL (a Seasonal-Trend decomposition procedure based on Loess) procedure (Cleveland et al., 1990), which is a filtering procedure for decomposing a time series into trend, seasonal, and remainder components with fixed seasonality. The amount of smoothing for the trend is taken to be the default in the *R* implementation. Otherwise, if the data is nonseasonal, the S_t term is set to 0, and the estimation of T_t is carried out using a penalized regression spline (Wood, 2000) with smoothing parameter chosen using cross validation;
- The transformation parameter λ is chosen to make the residuals from the decomposition as normal as possible in distribution that were often diagnosed as heterogeneous or non-Gaussian from the Diagnostic analysis. We choose $\lambda \in (-1, 1)$ to minimize

the Shapiro-Wilk statistic (Royston, 1982). We only consider a transformation if the minimum of $\{Y_t\}$ is non-negative. If the minimum of Y_t is zero, we add a small positive constant (equal to 0.001 of the maximum of Y_t) to all values to avoid undefined results.

3.2. Periodicity

Since the periodicity is very important for determining the seasonality and examining the cyclic pattern of the time series, the periodicity feature extraction becomes a necessity. Unfortunately, many time series available from the dataset in different domains do not always come with known frequency or regular periodicity like the series used in *M* competition, which was initiated by Spros Makridakis in 1982 with 1001 economic time series. In *M* competition data, time series which is monthly data has a period of 12 and daily data has a period of 7 or possibly 365. Therefore, we propose a new algorithm as follows to measure the periodicity in univariate time series. The periodicity detection is only applied for RAW data.

If there is a seasonal pattern of a certain period, the length of that time period can be used as an additional measure. Seasonal time series are sometimes also called periodic series although there is a major distinction between them. Periodicity (cyclical pattern) varies in frequency length, but seasonality has fixed length over each period. Periodicity is also referring as frequency in some literature. For time series with no seasonal pattern, the period is set to 1. We measure the periodicity using the following algorithm:

- Detrend time series using a regression spline with 3 knots
- Find $r_k = Corr(Y_t, Y_{t-k})$ (autocorrelation function) for all lags up to 1/3 of series length, then look for peaks and troughs in autocorrelation function.
- Frequency is the first peak provided with following conditions:
  (a) there is also a trough before it;
  (b) the difference between peak and trough is at least 0.1;
  (c) the peak corresponds to positive correlation.
- If no such peak is found, frequency is set to 1 (equivalent to non-seasonal).

3.3. Serial correlation

If a time series is white noise, $Y_t = c + E_t$, where c denotes an overall level of the sequence, and E_t is a random error component, which is uncorrelated from time to time. Therefore, we try to extract a measure which shows the degree of serial correlation of the dataset, to detect the series if it can fit a white noise model. Larger the degree is, more noisy the series is. Normally in the white noise series, there are no recurring cycles (periodicity) in the data because each observation is completely independent of all other observations. Two possible measures could be used which are described as follows:

- Autocorrelation (serial correlation): at a single time $r_k = Corr(Y_t, Y_{t-k})$, where k is the time lag;
- another measure called "Box-Pierce statistic" (Makridakis et al., 1998) which considers a whole set of r_k. Box-Pierce test Q_h, which was designed by Box and Pierce in 1970 for testing residuals from a forecast model (Box and Pierce, 1970), is also a common portmanteau tests for computing the measure. Box-Pierce statistic is $Q_h = n \sum_{k=1}^{h} r_k^2$, where n is the length of the time series, h is the maximum lag being considered (usually $h \approx 20$).

We have used Box-Pierce statistics in our approach to estimate the serial correlation measure, and to extract the measures from both RAW and TSA data.

3.4. Non-linear autoregressive structure

Nonlinear time series models have been used extensively in recent years to model complex dynamics not adequately represented use linear models (Harvill et al., 1999). For example, the well-known 'sunspot' datasets (Cleveland, 1994) and 'lynx' dataset (Hand et al., 1994) have identical non-linearity structure. Many economic time series are nonlinear when recession happens (Grossi and Riani, 2002). Because of the special characteristic (behavior) of time series data, the traditional linear models cannot handle the forecasting well compared to non-linear models. Therefore, non-linearity is an important characteristic of time series data to determine the selection of appropriate forecasting method.

There are many approaches to test the nonlinearity in time series regression models. Nonparametric kernel test and neural network test are the two major models appeared in the literature. In the comparative studies between these two approaches, neural network has been reported with better reliability (Lee, 2001). In this research, we used Teräesvirta's neural network test (Teräesvirta et al., 1993) for time series data nonlinearity characteristics identification and extraction. It has been widely accepted and reported that it can correctly model the nonlinear structure of the data (Rocca and Perna, 2004). It is a test for neglected nonlinearity likely to have power against a range of alternatives based on neural network model (augmented single-hidden-layer feedforward neural network model). The test is based on a test function chosen as the activations of 'phantom' hidden units. Refer to Teräesvirta (1996) for a detailed discussion on the testing procedures and formulas. We have used Teräesvirta's neural network test for nonlinearity (Teräesvirta et al., 1993).

3.5. Skewness

Skewness is a measure of symmetry, or more precisely, the lack of symmetry. A distribution, or dataset, is symmetric if it looks the same to the left and to the right of the center point. It is used to characterize the degree of asymmetry of values around the mean value.

For a univariate data Y_t, the skewness coefficient is: $S = \frac{1}{n\sigma^3} \sum_{t=1}^{n} (Y_t - \bar{Y}_t)^3$, where $\bar{Y}_t$ is the mean, σ is the standard deviation, and n is the number of data points.

The skewness for a normal distribution is zero, and any symmetric data should have the skewness near zero. Negative values for the skewness indicate data that are skewed left, and positive values for the skewness indicate data that are skewed right. In other words, left skewness means that the left tail is heavier than the right tail. Similarly, right skewness means the right tail is heavier than the left tail. Some measures have lower bounds that lead to skewness. For example, in reliability studies, failure times cannot be negative, so the data is skewed to the right.

3.6. Kurtosis (heavy-tails)

Kurtosis is a measure of whether the data are peaked or flat, relative to a normal distribution. A dataset with high kurtosis tends to have a distinct peak near the mean, decline rather rapidly, and have heavy tails. Datasets with low kurtosis tend to have a flat top near the mean rather than a sharp peak. For a univariate time series Y_t, the kurtosis coefficient is $K = \frac{1}{n\sigma^4} \sum_{t=1}^{n} (Y_t - \bar{Y}_t)^4$, where $\bar{Y}_t$ is the mean, σ is the standard deviation, and n is the number of data points. A uniform distribution would be the extreme case. The kurtosis for a standard normal distribution is 3. Therefore, the excess kurtosis is defined as $K = \frac{1}{n\sigma^4} \sum_{t=1}^{n} (Y_t - \bar{Y}_t)^4 - 3$. So, the standard normal distribution has a kurtosis of zero. Positive kurtosis indicates a "peaked" distribution and negative kurtosis indicates a "flat" distribution.

3.7. Self-similarity (Long-range dependence)

Processes with long-range dependence have attracted a good deal of attention from probabilists and theoretical physicists, and in 1984, Cox (1984) has presented a review of second-order statistical time series analysis. The subject of self-similarity and the estimation of statistical parameters of time series in the presence of long-range dependence are becoming more common in several fields of science (Rose, 1996), to which the time series analysis and forecasting on a recent research topic of network traffic, has drawn a particular attention. With such increasing importance of the 'self similarity (long-range dependence)' as one of time series characteristics, we decide to include this feature into proposed approach although it is not widely used or is neglected in time series feature identification.

The definition of self-similarity most related to the properties of time series is the self-similarity parameter Hurst exponent (H) (Willinger et al., 1996), and the details of the formulations is given in Rose (1996).

To estimate the Hurst parameter, traditional models such as ARMA and autoregressive integrated moving-average (ARIMA) are only capable of exploring short-range correlations of datasets. Therefore, the class of fractional autoregressive integrated moving-average (FARIMA) processes (Hosking, 1984) led from Brownian motion is a good estimation method for computing H. In a ARIMA (p, d, q), p is the order of AR, d is the degree first differencing involved, and q is the order of MA. If the time series is suspected to exhibit long-range dependency, d parameter may be replaced by certain non-integer values in FARIMA model. We fit a FARIMA (0, d, 0) by maximum likelihood which is approximated by using the fast and accurate method of Haslett and Raftery (1989). We then estimate the Hurst parameter using the relation $H = d + 0.5$. The Self-similarity feature is only detected from the RAW data.

3.8. Chaos (Dynamic systems)

Many systems in the nature that were previously considered random processes are now categorized as chaotic systems. Nonlinear dynamical systems often exhibit chaos, which is characterized by sensitive dependence on initial values, or more precisely by a positive Lyapunov Exponent (LE). Recognizing and quantifying chaos in time series represent important steps toward understanding the nature of random behavior and revealing the extent to which short-term forecasts may be improved (Lu, 1996).

LE as a measure of the divergence of nearby trajectories has been used to qualifying chaos by giving a quantitative value. The first algorithm of computing LE from time series was proposed by Wolf et al. (1985). It applies to continuous dynamical systems in an n-dimensional phase space. For a one-dimensional discrete time series, we used the method demonstrated by Hilborn (1994) to calculate LE of a one-dimensional time series (RAW data):

- Let Y_t denotes the time series;
- We consider the rate of divergence of nearby points in the series by looking at the trajectories of n periods ahead. Suppose Y_j and Y_i are two points in Y_t such that $|Y_j - Y_i|$ is small. Then we define $LE(Y_i, Y_j) = \frac{1}{n} \log \frac{|Y_{j+n} - Y_{i+n}|}{|Y_j - Y_i|}$;
- We estimate the LE of the series by averaging these values over all i values, choosing Y_j as the closest point to Y_i, where $i \neq j$. Thus, $LE = \frac{1}{N} \sum_{i=1}^{N} \lambda(Y_i, Y_i^*)$ where Y_i^* is the nearest point to Y_i.

3.9. Scaling transformations

The ranges of each of the above measures can vary significantly. In order to present the clustering algorithm with data rescaled in the [0, 1] range, so that certain features do not dominate the clustering, we perform a statistical transformation of the data. It is convenient to normalize variable ranges across a span of [0, 1]. Using anything less than the most convenient methods hardly contributes to easy and efficient completion of a task (Pyle, 1999). While we have experimented with linear and logistic transformations of the measures, we prefer the following more statistical approach. Three transformations (*f1*, *f2*, and *f3*) are used to rescale raw measure Q of different ranges to a value q in the [0, 1] range.

In order to map the raw measure Q of $[0, \infty)$ range to a rescaled value q in the [0, 1] rang, we use the transformation: $q = \frac{(e^{aQ} - 1)}{(b + e^{aQ})}$(referring as *f*1), where a and b are constants to be chosen. Similarly, for raw measure Q in the range [0, 1], we also use a transformation: $q = \frac{(e^{aQ} - 1)(b + e^{a})}{(b + e^{aQ})(e^{a} - 1)}$(referring as *f*2) to map to [0, 1], where a and b are constants to be chosen. We choose a and b such that q satisfies the conditions: q has 90th percentile of 0.10 when Y_t is standard normal white noise, and q has value of 0.9 for a well-known benchmark dataset with the required feature. For example, for measuring serial correlation, we use the Canadian Lynx dataset.

With raw measure Q in the $(1, \infty)$ range, for periodicity measure, we use another statistical transformation $q = \frac{(e^{\frac{(Q-a)}{b}} - 1)}{(1 + e^{\frac{(Q-a)}{b}})}$ (referring as *f*3), where a and b are constants to be chosen, with q satisfying the conditions: $q = 0.1$ for $Q = 12$ and $q = 0.9$ for $Q = 150$.

Table 2 Transformation parameters (transformation function, *a, b*) used in feature measures.

Feature	RAW data	TSA data
Serial correlation	*f*2, 7.53,0.103	*f*2,7.53,0.103
Non-linearity	*f*1,0.069,2.304	*f*1,0.069,2.304
Skewness	*f*1,1.510,5.993	*f*1,1.510,5.993
Kurtosis	*f*1,2.273,11567	*f*1,2.273,11567
Periodicity	*f*3,1,50	N/A

These frequencies ($Q = 12$ and $Q = 150$) were chosen as they allow the frequency range for real-world time series to fill the [0, 1] space.

For the measures that need rescaling, the transformation method and *a* and *b* values used in our measures extraction are listed in Table 2:

Apart from our scaling transformation, the other two popular scaling methods used in data process for data mining tasks are linear transform and softmax scaling methods (which will be used for comparison in clustering experiments later in the paper):

- A linear transform method, mapping $(-\infty, \infty)$ to [0, 1] range: $v_{\text{norm}} = \frac{v_i - \min(v_1 \ldots v_n)}{\max(v_1 \ldots v_n) - \min(v_1 \ldots v_n)}$, where v_{norm} is the normalized value and v_i is a instance value of actual values;
- The Softmax scaling (the logistic function) to map $(-\infty, \infty)$ to (0, 1): $v_n = \frac{1}{1+e^{-v_i}}$, where $e^{-v_i} = \frac{1}{e^{v_i}}$, v_n denotes the normalized value and v_i denotes a instance value of actual values.

Now that the global characteristic features have been defined, we then have a means of extracting the basic measures from a time series. Using this finite set of measures to characterize the time series, regardless of their length or missing data, the time series datasets can be clustered using any appropriate clustering algorithms. In the following section we describe the clustering process and experimental result to demonstrate the reliability of the proposed method.

4. Clustering

In this paper, we restrict our attention to hierarchical clustering and self organizing map clustering for their advantage in visualization.

4.1. Hierarchical clustering

Hierarchical clustering algorithm is a well-known clustering method which has been applied in many domains. In visualizing the result, a dendrogram is generated from the clustering process, representing the nested grouping of patterns and similarity levels at which groupings change. There are three major variants of hierarchical clustering algorithms. They are Single-link, complete-link, and minimum-variance algorithms. Of these three, the single-link and complete-link algorithms are most popular; more details can be found in Jain et al. (1999).

4.2. *Self organizing map clustering*

The SOM is both a projection method which maps high-dimensional data space into low-dimensional space, and a clustering method so that similar data samples tend to map to nearby neurons. Since its introduction in the early 1980s (Kohonen et al., 2002), the SOM has widely been adopted as a statistical tool for multivariate analysis (Honkela, 1997). We have chosen to include SOM's for clustering in our approach due to its robustness in parameter selection, natural clustering results, and superior visualization compared to other clustering methods, such as hierarchical and *K*-means.

4.3. *Empirical evaluation*

The set of feature measures extracted from each time series in a dataset are ready to form the input vector for both clustering algorithms directly; simply, there is no further data pre-processing required.

To provide a convincing empirical evaluation, we performed the experiments on various datasets, and for comparison purpose, three widely used benchmark datasets were selected. Two datasets from the UCR Time Series Data Mining Archive (Keogh and Folias, 2002), "reality check" and "18 pairs," have been tested for clustering by other researchers and used as benchmarking for comparison. "Reality check" consists of data from space shuttle telemetry, exchange rates, and artificial sequences. The data is normalized so that the minimum value is zero and the maximum is one. There are fourteen time series and each contains 1,000 data points. In "18 pairs" dataset, thirty-six time series with 1,000 data points each come in eighteen pairs. In addition, we also included a dataset with "known features", such as seasonal data, chaotic data, etc., consisting of both real series and synthetic series to demonstrate the method's capability. There are twenty-four time series in this dataset; the maximum length of the time series in this dataset is also 1,000 data points.

4.3.1. Experiments with hierarchical clustering. We have chosen the complete-link hierarchical clustering algorithm to achieve more useful hierarchies than single-link from a pragmatic viewpoint (Jain et al., 1999). In the experiments, the extracted features of time series in each dataset have been used as clustering inputs. Dendrograms were created to help ease the visualization of clusters and clusters' relationship.

In order to compare the results with the benchmarked clusters generated by hierarchical clustering of the raw time series data (Keogh and Folias, 2002), we have clustered the data series in three benchmark datasets based on normalized thirteen extracted statistical feature measures using hierarchical clustering method. As shown in Figures 1–3, our global features, only 13 measures of the time series, are adequate to generate fairly good clusters from hierarchical clustering. The hierarchical clustering based on extracted measures was able to group the similar pattern closer together, as indicated with same shade of the series in Figures 1 and 3.

4.3.2. Experiments with SOM clustering. SOM as a clustering technique has been used for dimension reduction in data mining tasks for high dimensional time series. SOM has been used before for time series clustering (Debregeas and Hebrail, 1998; Van Laerhoven, 2001) to project the time series onto a 2-D space to visualize the clustering

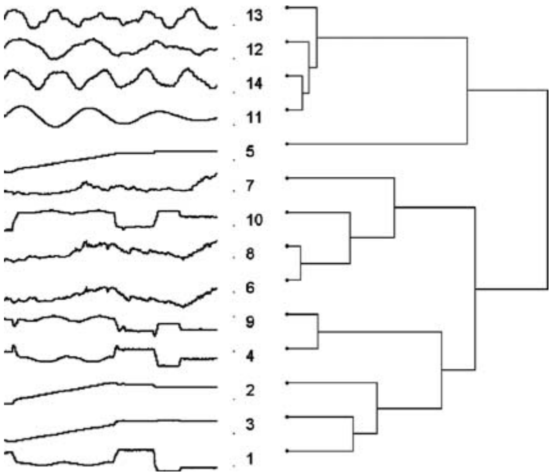

Figure 1 Dendrogram from hierarchical clustering on "reality check" dataset.

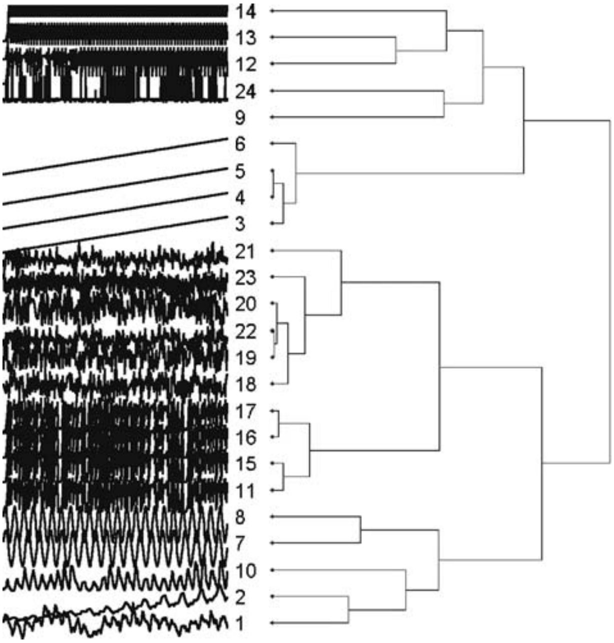

Figure 2 Dendrogram from hierarchical clustering on "known feature" dataset.

result. The feature measures extracted from the time series, after applying the statistical transformation, normalizing into (0, 1) range, have been fed into SOM for clustering process. We have used the Viscovery SOMine (see www.eudaptics.at) to generate a 2-D map. A trial-and-error process was used to determine a few parameter settings by comparing the normalized distortion error and quantization error. In our experiments, it normally took around 70 iterations before the parameters to be determined. The final

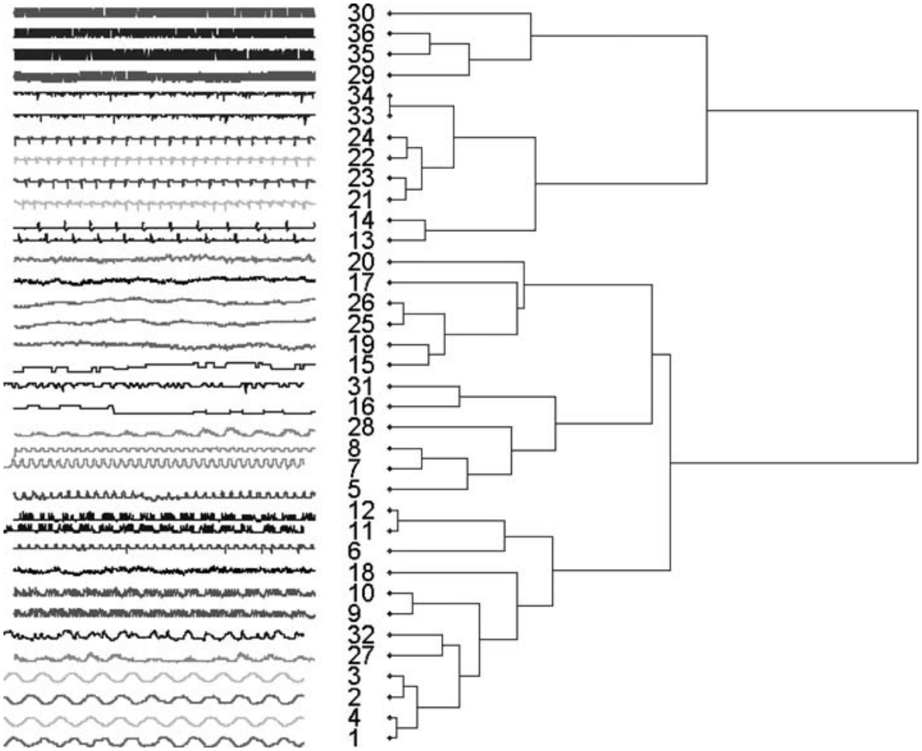

Figure 3 Dendrogram from hierarchical clustering on "18 pairs" dataset.

maps were generated after achieving the minimum errors for both normalized distortion and quantization. The distance between datasets on the map indicates the similarity between them. To illustrate the clustering robustness and feature identifying capability, we have plotted the 2-D map of "known features" dataset, as shown in Figure 4. From the allocation of each time series with specific feature, we can see that the clusters are stabilized with same feature identified.

If we examine the data distribution (or location) in the clustered 2-D map geometrically, this map with time series can be circled with different zooms as in the real-world city map. As illustrated in Figure 4, three zooms have been circuited in ring shape in this 2-D map. The linear time series examples of perfect trend lines have been grouped together and only appear in Zoom 1 (on the left bottom corner of the map). When the radius of the circle increases, more series are located in the next area, identified as Zoom 2 in the map. The time series include seasonal, self-similarity and chaotic time series which are clustered together and immediately outside the linear perfect trend series of Zoom 1 on the map. We can see that the nonlinearity feature is increasing when the map is zooming out, which is confirmed by checking the time series in Zoom 3 (the most outside or the right border part of the map). All the time series in Zoom 3 have strong nonlinearity, such as random data and chaotic with noise data. By utilizing the visualization advantage of SOM, it is clear that our clustering method, CBC, is able to yield more meaningful

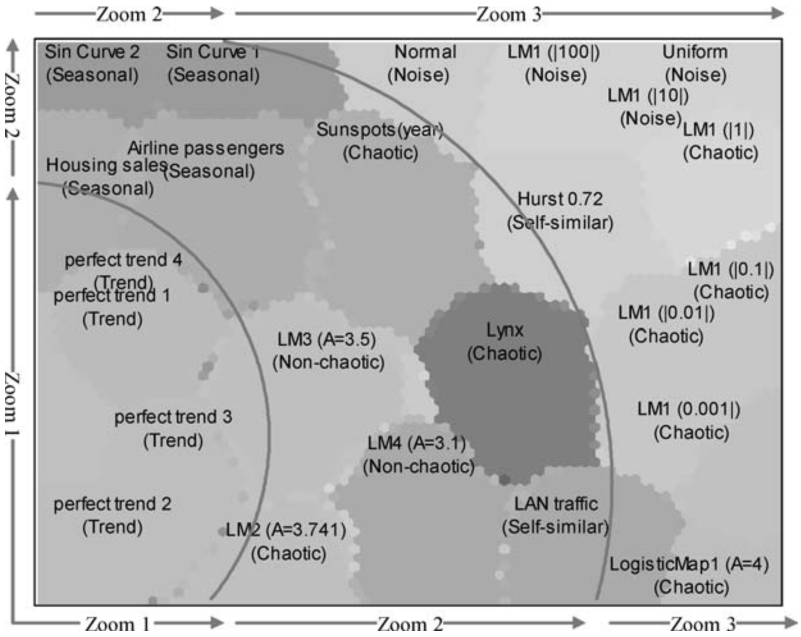

Figure 4 A 2-D SOM map showing clusters on "known features" dataset.

clustering results for time series data by taking both global and local cluster relationships into consideration.

4.3.3. Comparison with related work. When benchmark datasets are used, it is perhaps more interesting to see how the results compare to those obtained by other researchers. Therefore, we choose one benchmark dataset to evaluate and compare the clustering result based on our approach to other research's outcome.

Using the same "reality check" dataset, Keogh has produced a dendrogram (Keogh et al., 2003) using hierarchical clustering structure in which the actual data points are inputs for clustering process (as shown in Figure 5). In the experiments using SOM to generate clusters, we have tuned the number of clusters in the range of [2, 14] as the cluster numbers acquired from hierarchical clustering experiment. Therefore, we were able to re-interpret the clusters generated by the SOM for the same dataset into a hierarchical structure (as shown in Figure 6) for a visualized comparison with Keogh's result. Comparing the dendrograms in both figures, we can readily see that similar clusters have been obtained from our approach. Our clustering results with SOM for clustering are arguably better, or at least more intuitive. For example, in Figure 5, series 1 and 4 and series 9 and 10 have been grouped far from each other based on the hierarchical clustering using actual data point values. On the other hand, the result in Figure 6 was generated by using our global characteristic measures as inputs, they have been grouped together in the dendrogram. Through a visual inspection of these series, it does show that they are actually quite similar in character. It indicates that using our proposed global measures, the clustering algorithm is aware of the "whole picture" and is able to recognize the similarity of these four time series among other series in the dataset. In

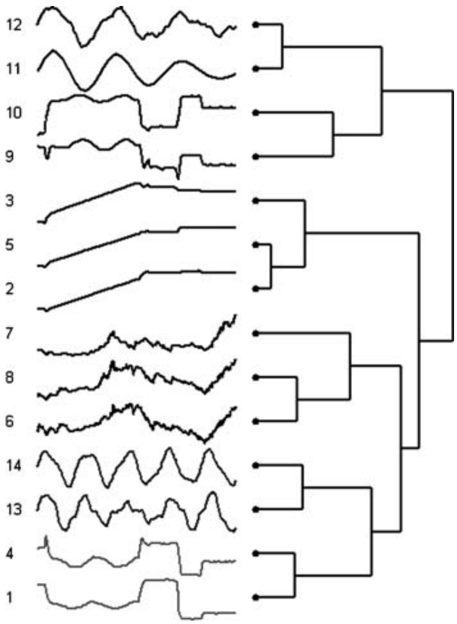

Figure 5 Dendrogram from hierarchical clustering using actual points on "reality check" dataset (Keogh et al., 2003).

addition to identifying clusters, as illustrated in Figure 6, our method performed better in detecting the global characteristics of the time series (from highly seasonal at the top to linear trend at the bottom). The result shows that our method is able to provide good clusters in the hierarchical tree structure. In other words, the same accuracy has been achieved in the lower level, and a better relationship has been represented in the higher level.

In a comparative study of quality of clustering using different methods (including clustering inputs and distance measures) (Keogh et al., 2004), CDM approach was reported as the best among others. In their study, the "18 pairs" dataset was used benchmark dataset to produce dendrogram from hierarchical clustering. A metric, *PU*,

Table 3 Clustering performance comparison.

Methods	PU
Compression-based dissimilarity measure	1
Characteristic-based clustering using all features without optimization	*0.55*
Dynamic time warping	0.33
Longest common subsequence	0.33
Hidden markov model + Symbolic aggregate approximation	0.33
Euclidean distance	0.27
Autocorrelation method	0.16
Hidden markov model	0

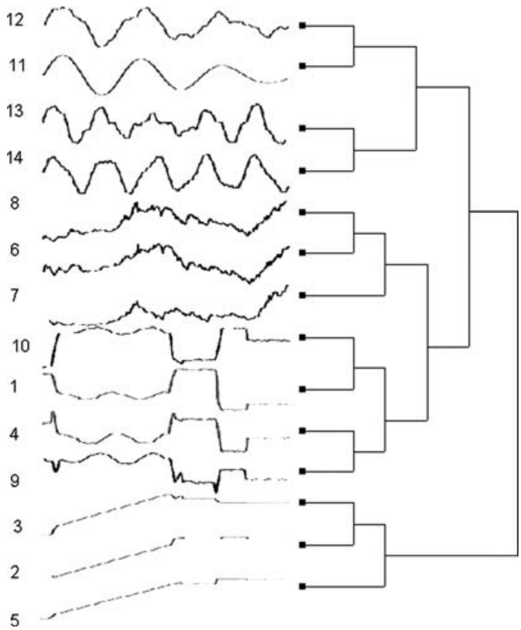

Figure 6 Dendrogram converted from SOM clusters on “reality check” dataset.

the quality of clustering is measured as the number of correct bifurcations divided by eighteen because the known 18-pair. Referring to the better performing measures in Keogh et al. (2004), our CBC approach has achieved a promising accuracy compared with others, as listed in Table 3. From this, we gained a confidence in the capability of our CBC, and in later section, we will further tune our method to improve its performance through a search mechanism.

The series in these tested datasets consist of data ranging from a synthetic “perfect trend line” to real-world chaotic data set “lynx.” The lengths of these time series are varying from 300 to 30,000 data points and the data for the clustering process does not need to be pre-processed. The empirical result clearly demonstrated that our method is robust for clustering of different-length time series, using a finite set of global characteristics measures.

4.3.4. Tests on scaling methods. As mentioned above, because there is no further data pre-processing required after the measures are extracted from a time series, they can directly be used as inputs for clustering algorithms. However, there is a scaling operation at the last extraction process through scaling transformation, which may produce different final clustering result.

Therefore, we have tested the robustness of the proposed approach using different scaling methods described in Section 2.9, and the significance of the different measures were evaluated by generating numerous clusters using different variations of inputs and data transformations. We have found that different scaling transformations on the measures did not significantly affect the resulting map, which shows that our approach is not sensitive to data scaling in preparation.

5. Feature searching

In the previous section of the paper, the clustering based on extracted global characteristics measures has empirically shown the ability in robust clustering on some benchmark datasets. While the reducibility of the proposed approach should be apparent from the fact that we are extracting a finite set of features, it is most important to determine if the selected measures of global characteristics can generate high-quality clustering. If a good set of features are extracted, data mining algorithms will not only be speeded up, they will produce even better results than applying the algorithm on the original data. In particular, clustering algorithms will profit from the data preprocessing (Mörchen, 2003).

In practice, the question of generality of the method always arises; how well would our identified features generalize to other datasets and domains. We have realized that a different selection from our limited number of global measures could affect the clustering result. To answer this question and generalize the clustering approach, a search mechanism is designed to optimize the feature set for different types of application domains.

5.1. Forward search algorithm

Our problem is to find the best set of *n* features from our set of *m* possible features. As noted above, *m* is currently 13, although this number could be increased in future as new feature extraction methods become available. There are $2^{13} - 1 = 8{,}191$ possible subsets of 13 measures, clearly testing each one is untenable. To make the feature subset selection tractable we will use a greedy Forward Search (FS). Forward search is a powerful general method for detecting multiple influences in a model (Atkinson and Riani, 2000). FS is only optimal for models which assumed independent observations: i.e linear and non linear regression, generalized linear models and multivariate analysis (Grossi and Riani, 2002), however even in cases where the operators are not independent, it has been shown to be very robust in practice.

Figure 7 illustrates the intuition behind FS. We begin from an empty set of features, and consider adding each of the 13 features one by one. Using some measure of quality (discussed below), we find the best single feature to add the feature set. After selecting this feature, we then individually consider every adding each of the remaining features; we continue this process until *all* features have been added (to guard against local minima). After all features have been added we select the feature subset that gave the maximum value for the measure of quality.

We need to consider what measure of quality to use as a heuristic to guide the search process. We consider two possibilities, supervised and unsupervised. Using a supervised measure is only *apparently* in conflict with the goal of clustering. The idea is to do the search on a small subset of the data, for which labels are known, in order to learn the feature subset that can then be applied to a larger unlabeled dataset. Using labeled data we use simple one nearest neighbor classification as the quality measure.

For the unlabeled case we use a simple heuristic based on clustering stability. If a feature were strongly correlated with the underlying latent "classes" in the data, we would expect that the clusterings produced using that feature would be fairly robust to small perturbances in the value of that feature. So we run multiple restarts of *K*-means

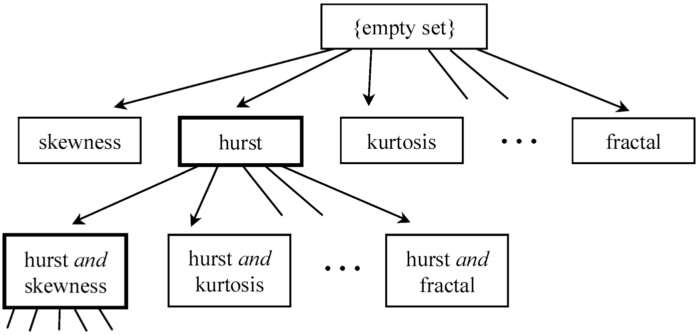

Figure 7 A sketch of the forward search algorithm.

clustering with the values of the feature in question randomly changed by 1% to10%. We can then use the stability of clustering (as defined in Lange et al. (2004)) as a measure of feature quality.

5.2. *Empirical evaluation*

To provide a convincing empirical evaluation, we test our technique on both synthetic and real-world datasets. A synthetic dataset called '2-class' consists of 40 series with 5,000 data points in each, which include twenty 'random walk' series and twenty 'shuttle' series for both training set and test set. To tune the measures for time series in different domains, another real-world dataset 'eeg' dataset with 5 classes has been tested in greedy search for demonstration.

5.2.1. Evaluation on a 2-class problem with synthetic dataset. Using the synthetic dataset, we employed our search tool to find the best feature measures for training set, and then only the selected feature measures are used for clustering the test set. We wish to determine whether the outcome from search can produce a good clustering and how it affects the clustering results.

A search was performed on the training set with known classes, and then the features are listed in order of utility for the clustering accuracy, as illustrated in Table 4 and Figure 8.

Using the knowledge gleaned from the search result in feature measure selection on training set, we use two subsets from the feature measure set to evaluate the clustering on test set. From the total thirteen feature measures, Subset I is extracted which only includes three measures (serialcorrelation-raw, skewness-tsa, and hurst-raw) given 92.5% clustering accuracy which is the highest point before accuracy decrease (Table 4). Subset II involves ten measures (serialcorrelation-raw, skewness-tsa, hurst-raw, seasonal-tsa, periodicity-raw, skewness-raw, serialcorrelation-tsa, trend-tsa, kurtosis-tsa, fractal-raw, and nonlinear-raw) which give 90.0% clustering accuracy (the second highest point before drop). An empirical comparison was implemented using different inputs on the test set. As illustrated in Table 5, the number of instances, that were correctly clustered using three different selections from the proposed 13 measures, is higher than using actual data points as inputs for the clustering. To exam and compare the quality of the

Table 4 Classification accuracy for feature measures input.

Feature measures	Accuracy (%)
SerialCorrelation-raw by itself	90
Skewness-tsa with the above	92.5
Hurst-raw with the above	92.5 (subset I)
Seasonal-tsa with the above	90
Periodicity-raw with the above	87.5
Skewness-raw with the above	87.5
SerialCorrelation-tsa with the above	87.5
Trend-tsa with the above	87.5
Kurtosis-tsa with the above	85
Fractal-raw with the above	90
Nonlinear-raw with the above	90 (subset II)
Kurtosis-raw with the above	87.5
Nonlinear-tsa with the above	82.5

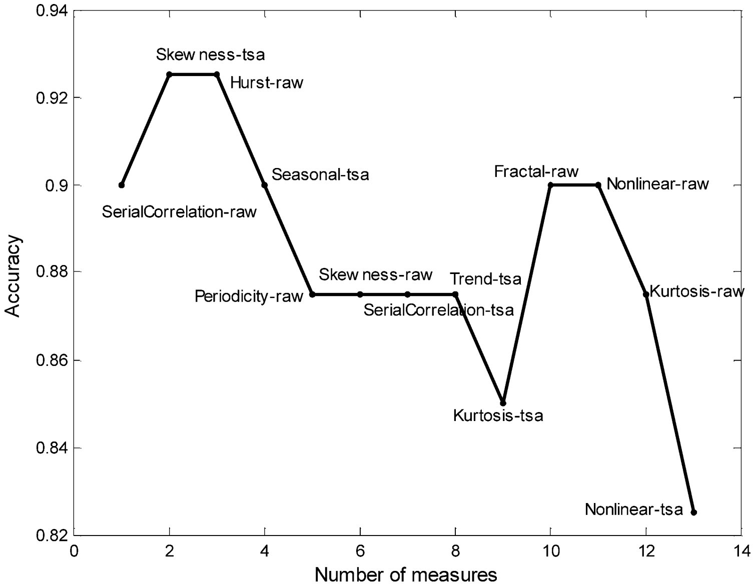

Figure 8 Classification accuracy changes as features are added.

clustering results on the test set, we can take the correctly clustered number of instance only from each class, and divided by 20 (which is the known number of series in each class). Then the clustering results can be compared using one simple figure as shown in Table 6. Retrieving the subsets selection based on the classification accuracy (in Table 4), subset II has lower accuracy than subset I, which means the subset I should produce better clustering than subset II. It has been verified in the experiments results

Table 5 Clustering evaluation on Testing Dataset using different selection of inputs.

	Class 1		Class 2	
Inputs	Correctly clustered	Wrongly clustered	Correctly clustered	Wrong clustered
all actual data points	8	0	20	12
13 extracted measures	19	1	19	1
3 measures (subset I)	14	3	17	6
11 measures (subset II)	10	11	9	0

Table 6 Clustering evaluation on Testing Dataset using different selection of inputs.

Inputs	Class 1	Class 2
all actual data points	40%	100%
13 extracted measures	95%	95%
3 measures (subset I)	70%	85%
11 measures (subset II)	50%	45%

the test set, and it also shows that the search mechanism produced useful information for measures selection.

To provide a simple visual examination and comparison among the clustering results, we generated the dendrograms using all actual data points as input (which is plotted in Figure 9) and proposed feature measures as input (as in Figure 10). The time series labeled [1:20] were annotated as random walk (class 1) and time series labeled [21:40] were annotated as 'shuttle' (class 2). It is clear to see that using the feature measures sets as inputs for clustering produced higher quality results than the actual data points. Such result shows that excessive number of inputs (or larger number of inputs) is not necessary to generate better clustering result compared to less number of inputs if the less number of inputs carry more meaningful information (characteristics) of the time series to be clustered.

5.2.2. Evaluation on a 5-class problem with real-world dataset. A central claim of this work is that feature sets that are found (by search) to be useful on a particular dataset from a particular domain are very likely to generalize to future datasets from the same domain. Below we will test this hypothesis. The basic idea of this experiment is to use our search algorithm to learn a good feature set from all available features on an annotated dataset of electrocardiograms, and see how well this feature set can cluster unseen data.

The problem is a 5-class problem (as labeled in 'Z','N','S','O'and 'F'), with the annotations being provided by a cardiologist. Both the training and test datasets have 50 examples in each class. Each example is a time series with 4097 observation data points.

We used all 13-feature measures (called the "wholeset") as operators for our search algorithm, and we used classification accuracy as the search criteria, the accuracy when each measure adding in is listed in Table 7 and graphed in Figure 11.

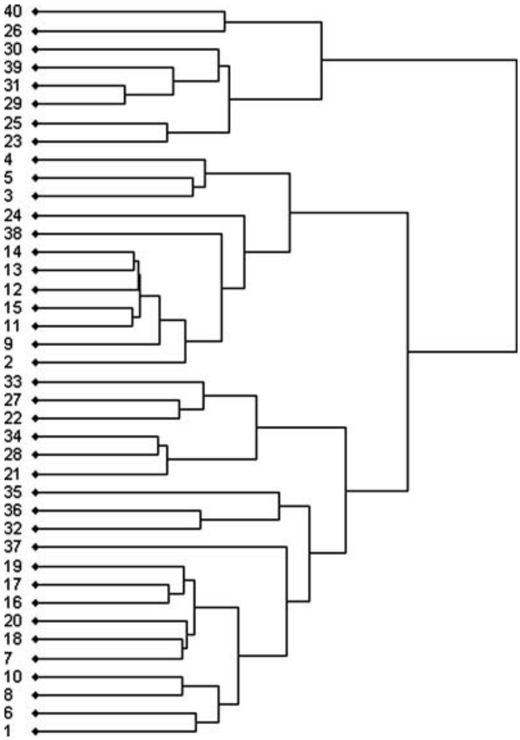

Figure 9 Dendrogram from hierarchical clustering using actual data points.

Table 7 Classification accuracy for feature measures input.

Feature measures	Accuracy (%)
Frequency by itself	48.80
Lyapunov with the above	63.60
Serial Correlation-DC with the above	70.40
Trend-DC with the above	74.40 (F4)
Skewness-DC with the above	76
Hurst with the above	76
Seasonal-DC with the above	76 (F7)
Nonlinear-DC with the above	74.80
Kurtosis-DC with the above	76.40
Nonlinearity with the above	76
Serial Correlation with the above	76.80
Skewness with the above	77.20 (F12)
Kurtosis with the above	75.20 (F13)

Table 8 Clustering count (accuracy) on wholeset F13.

	Class Z	Class N	Class S	Class O	Class F	Mean
Hierarchical clustering	2.0%	12.0%	26.0%	100.0%	4.0%	*28.8%*
Self organizing map	0.0%	26.0%	78.0%	66.0%	12.0%	*36.4%*
K-Means clustering	68.0%	14.0%	38.0%	98.0%	16.0%	*46.8%*
Fuzzy cmeans clustering	68.0%	16.0%	38.0%	98.0%	16.0%	*47.2%*
Mean	*34.5%*	*17.0%*	*45.0%*	*90.5%*	*12.0%*	*39.8%*

While the accuracy of classification peaks with the addition of the 12th feature (skewness), a visual inspection strongly suggests that this is a case of overfitting. The accuracy with 12 features is only marginally better than that with first four features. The idea that we should penalize *apparent* increases in accuracy produced as a result of a more complex model (Wallace, 1999), or as a result of oversearching (Domingos, 1999) is well understood in the data mining and statistical communities. We could potentially leverage such frameworks to produce automatic techniques to determine the *number* of features to keep, however for simplicity here we content ourselves with visual inspection.

We tested the 4 following possibilities on the task of clustering the training set.

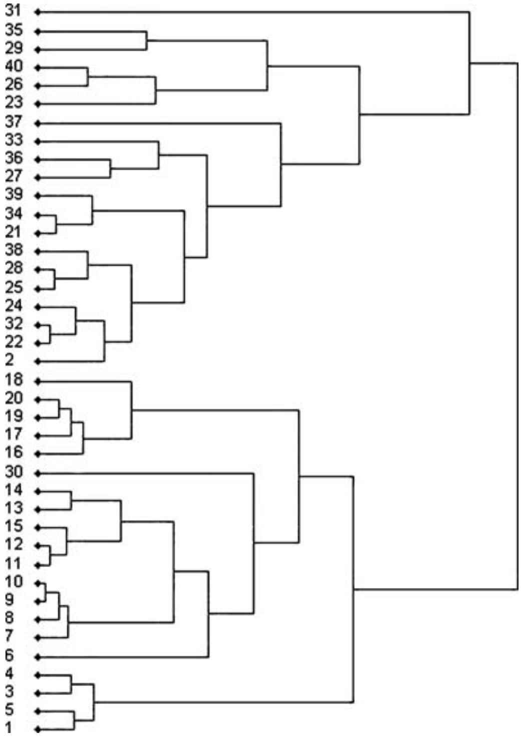

Figure 10 Dendrogram from hierarchical clustering using global measures.

Table 9 Clustering count (accuracy) on subset F12.

	Class Z	Class N	Class S	Class O	Class F	Mean
Hierarchical clustering	0.0%	12.0%	52.0%	100.0%	4.0%	*33.6%*
Self organizing map	92.0%	74.0%	70.0%	0.0%	34.0%	*54.0%*
K-means clustering	0.0%	66.0%	48.0%	100.0%	12.0%	*45.2%*
Fuzzy cmeans clustering	92.0%	66.0%	46.0%	0.0%	14.0%	*43.6%*
Mean	*46.0%*	*54.5%*	*54.0%*	*50.0%*	*16.0%*	*44.1%*

Table 10 Clustering count (accuracy) on subset F4.

	Class Z	Class N	Class S	Class O	Class F	Mean
Hierarchical clustering	92.0%	78.0%	56.0%	0.0%	10.0%	*47.2%*
Self organizing map	70.0%	78.0%	68.0%	0.0%	14.0%	*46.0%*
K-Means clustering	92.0%	70.0%	54.0%	64.0%	0.0%	*56.0%*
Fuzzy cmeans clustering	88.0%	78.0%	46.0%	58.0%	14.0%	*56.8%*
Mean	*85.5%*	*76.0%*	*56.0%*	*30.5%*	*9.5%*	*51.5%*

Table 11 Clustering count (accuracy) on subset F7.

	Class Z	Class N	Class S	Class O	Class F	Mean
Hierarchical clustering	92.0%	78.0%	48.0%	0.0%	10.0%	*45.6%*
Self organizing map	0.0%	74.0%	42.0%	100.0%	40.0%	*51.2%*
K-Means clustering	92.0%	74.0%	56.0%	64.0%	6.0%	*58.4%*
Fuzzy cmeans clustering	92.0%	76.0%	56.0%	64.0%	6.0%	*58.8%*
Mean	*69.0%*	*75.5%*	*50.5%*	*57.0%*	*15.5%*	*53.5%*

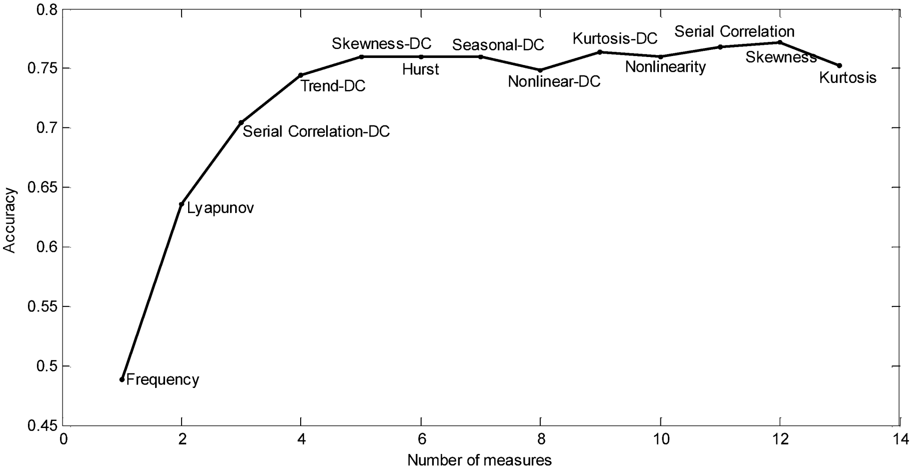

Figure 11 Classification accuracy changes as features are added.

- F13: All 13 features, this is the naïve "the more features the better" approach.
- F12: This is the highest accuracy achieved during search on training set.
- F7: This is highest value obtained before a decrease was observed.
- F4: Unlike this above 3 possibilities, this choice (of the first four features) is subjective. We noted that adding the forth feature gives us more the 90% of the eventual improvement, but that we have double the number of features to see any further positive change.

We tested four different clustering algorithms: SOM (Viscovery SOMine software), *K*-means and Fuzzy-cmeans (David Corney's clustering toolbox for matlab) and complete-linkage hierarchical clustering. The latter does not directly produce a partitional clustering, but can be converted to partitions by the commonly used trick of cutting the trees *k* top-most branches and regarding the sub-trees as discrete clusters.

We measure the accuracy of each clustering algorithm by averaging the cluster purity of the five resulting clusters. The cluster purity is defined as the number of the dominant class with that cluster, divided by 50 (since 50 is the known number of each class). For example, if one cluster contains 10 objects labeled {Z, Z, Z, Z, Z, S, N, O, F, F}, then the class *Z* dominates with five occurrences and the cluster purity is 5/50 $=$ 10%. When the number of final clusters is five, we recognize each cluster with one dominative class from five known classes (as labeled in 'Z','N','S','O'and 'F'). We do not allow the same class to dominate more than one resulting cluster although its clustering count could be very high in more than one cluster. Then, we can acquire the clustering count for each of five classes based on five clusters obtained from the experiments. This measure ranges from 100% for a perfect clustering to approximately 20% for random clustering (regardless of the distribution of cluster sizes). The Tables 8–11 give the results of the experiments using different selection of inputs and various clustering algorithms.

In general the results of all four algorithms are similar, but *K*-means and Fuzzy-cmeans produced better clusterings compared to two algorithms (hierarchical and SOM). In our experiments, SOM was fairly better than Hierarchical clustering in most cases. Let us consider the average of the performance from both algorithms and classes perspective, it is very clear that using all 13 features gives the lowest accuracy (39.8%), only is not greatly better than random chance (20%). If we use 12 features, the recognition of distinguished classes has been improved and overall accuracy also increased to 44.1%. Finally, when we used F7 (the highest value obtained before a decrease was observed in feature selection using training set), the highest clustering accuracy on the test set also reached. This information provided us the verification of the expected best selection of subset found by forward search mechanism. Using seven features gave us the best accuracy of 53.5%, while our more conservative feature set (F4) of the four best features also gave us great accuracy of 51.5%. It tells that although F12 and F7 are the subsets selection with highest classification accuracies during the search against the clustering results on training set, they are not necessary the best choice as the features selection, which we had anticipated with our discussion of overfitting and oversearching.

These results strongly support our claim that the results learned on one dataset will generalize to future datasets of the same type.

5.2.3. Computational complexity consideration. The computational time for calculating all 13 features is very fast due to their linear or logarithmic complexities. We have tested the computational complexity (in system CPU time) for different type of data in various lengths ranging from 500 to 10000 observations in each series. The testing with our *R* code to extract the feature measures was run on the PC with the system specifications as: Intel Celeron (R) processor, 2.0 Ghz CPU, 20 GB HDD, 512 MB of RAM, Windows XP. From over fifty times experiments, the results are between 0.5 to 2 seconds for each feature.

From both theoretical and practical perspectives, most clustering algorithms can work more efficiently with less number of inputs. As such, our approach, CBC, is much faster than other ordinary clustering methods in computation. Because it only uses a small number of inputs as low dimensional input vector in the clustering process, but others still have to work on actual data points which is often in high dimensionality (or with large number of data inputs).

The computational complexity for the forward search used in our approach has a simple linear complexity, $O(s, n, n')$, in which s is the number of base classifiers, n is the total number of features, and n' is the number of features included in the searching. Because we have a small size of features and the number of classifiers depends on the problem domain which is usually a small number as well. Therefore, the forward search to select the optimal set of features is inexpensive in computation.

Although the feature extraction and forward search are additional processes involved in CBC approach compared to general standard clustering, the computational complexity is not a negative factor. However, the proposed CBC method for time series clustering is able to provide robust results with less complexity.

6. Conclusions

In this paper we have proposed a new general framework for time series clustering. We evaluated our method on a collection of benchmark time series clustering data. Our empirical results demonstrate that our proposed characteristic-based clustering is able to cluster time series using just a set of derived features. Using only a relatively small set of global measures, we can still achieve clustering with high accuracy. The knowledge provided to the clustering algorithm by the global measures appears to benefit the quality of the clustering results.

The main advantage of our approach is its ability to drastically reduce the dimensionality of the original time series, which overcomes a severe limitation of most of the existing approaches to time series clustering. In addition, by working at a higher level abstraction of the data, our approach is much less sensitive to missing values. As an additional advantage, our approach only requires the setting of the parameters of the selected clustering algorithm and transformation parameters; no additional parameters are introduced by the feature extraction process.

An appropriate characterization of time series data can reveal the relationship between features and clustering accuracy. Our current work uses a small set of commonly used features with the most popular clustering algorithms. Although only few parameters are involved in the proposed method, the sensitivity of the selection of these parameters (for example, parameters a and b in the transformation) has not been investigated thoroughly regarding the clustering results. In the future, we plan to examine the current method in more details and investigate more advanced features and clustering algorithms. We also intend to explore how the metrics may change over time for chronological clustering, and identifying dynamic changes in time series.

Acknowledgment

The first author is grateful to Monash University for providing a Postgraduate Publication Award to support the manuscript preparation.

References

Agrawal, R., Faloutsos, C., and Swami, A. 1993. Efficient similarity search in sequence databases. In Proc. of the 4th International Conference on Foundations of Data Organization and Algorithms, Chicago, IL, USA, pp. 69–84.

Armstrong, J.S. (Ed.), 2001. Principles of Forecasting: A Handbook for Researchers and Practitioners. Kluwer Academic Publishers.

Atkinson, A.C. and Riani, M. 2000. Robust Diagnostic Regression Analysis, New York: Springer.

Berndt, D. and Clifford, J. 1994. Using dynamic time warping to find patterns in time series. In Proc. of the AAAI'94 Workshop on Knowledge Discovery in Databases, pp. 229–248.

Box, G.E.P. and Cox, D.R. 1964. An analysis of transformations. JRSS, B(26):211–246.

Box, G.E.P. and Pierce, D.A. 1970. Distribution of the residual autocorrelations in autoregressive-integrated moving-average time series models. Journal of the American Statistical Association, 65:1509–1526.

Bradley, P.S. and Fayyad, U.M. 1998. Refining initial points for k-means clustering. In Proc. of the 15th International Conference on Machine Learning, Madison, WI, USA, pp. 91–99.

Chan, K. and Fu, A.W. 1999. Efficient time series matching by wavelets. In Proc. of the 15th IEEE International conference on data engineering, Sydney, Australia, pp. 126–133.

Chatfield, C. 1996. The Analysis of Time Series: An Introduction. London: Chapman & Hall.

Chu, K. and Wong, M. 1999. Fast time-series searching with scaling and shifting. In Proc. of the 18th ACM Symposium on Principles of Database Systems, Philadelphia, PA, USA, pp. 237–248.

Cleveland, R.B., Cleveland, W.S., McRae, J.E., and Terpenning, I. 1990. Stl: A seasonal-trend decomposition procedure based on loess. Journal of Official Statistics, 6:3 –73.

Cleveland, W.S. 1994. The Elements of Graphing Data. NJ: Hobart Press Summit.

Cox, D.R. 1984. Long-range dependence: A review. In Proc. of the Statistics: An Appraisal, 50th Anniversary Conference, Iowa State Statistical Laboratory, pp. 55–74.

Debregeas, A. and Hebrail, G. 1998. Interactive interpretation of kohonen maps applied to curves. In Proc. of the 4th International Conference of Knowledge Discovery and Data Mining, New York, NY, USA, pp. 179–183.

Dellaert, F.T. Polzin, T., and Waibel, A. 1996. Recognizing emotion in speech. In Proc. of the 4th International Conference on Spoken Language Processing, Philadelphia, PA, USA, pp. 1970–1973.

Deng, K. Moore, A. and Nechyba, M.C. 1997. Learning to recognize time series: Combining arma models with memory-based learning. In Proc. of the International Symposium on Computational Intelligence in Robotics and Automation, pp. 246–50.

Domingos, P. 1999. Role of occam's razor in knowledge discovery. Data Mining and Knowledge Discovery, 3:409–425.

Faloutsos, C., Ranganathan, M., and Manolopoulos, Y. 1994. Fast subsequence matching in time-series databases. In Proc. of the ACM SIGMOD International Conference on Management of Data, Minneapolis, MN, USA, pp. 419–429.

Ge, X. and Smyth, P. 2000. Deformable markov model templates for time-series pattern matching. In Proc. of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Boston, Massachusetts, pp. 81–90.

Grossi, L. and Riani, M. 2002. Robust time series analysis through the forward search. In Proc. of the 15th Symposium of Computational Statistics, Berlin, Germany, pp. 521–526.

Halkidi, M., Batistakis, Y., and Vazirgiannis, M. 2001. On clustering validation techniques. Journal of Intelligent Information Systems (JIIS), 17(2/3):107–145.

Hamilton, J.D. 1994. Time Series Analysis. Princeton University Press, Princeton.

Hand, D.J., Daly, F., Lunn, A.D., McConway, K.J., and Ostrowski, E. 1994. A Handbook of Small Data Sets. Chapman & Hall, London.

Harvill, J.L., Ray, B.K., and Harvill, J.L. 1999. Testing for nonlinearity in a vector time series. Biometrika, 86:728–734.

Haslett, J. and Raftery, A.E. 1989. Space-time modelling with long-memory dependence: Assessing ireland's wind power resource (with discussion). Applied Statistics, 38:1–50.

Hilborn, R.C. 1994. Chaos and Nonlinear Dynamics: An Introduction for Scientists and Engineers. Oxford University Press, New York.

Honkela, T. 1997. Self-Organizing maps in natural language processing, Ph.D. Thesis, Neural Networks Research Centre, Helsinki University of Technology.

Hosking, J.R.M. 1984. Modeling persistence in hydrological time series using fractional differencing. Water Resources Research, 20(12):1898–1908.

Huntala, Y., Karkkainen, J., and Toivonen, H. 1999. Mining for similarities in aligned time series using wavelets. In Proc. of the Data Mining and Knowledge Discovery: Theory, Tools, and Technology, Orlando, FL, pp. 150–160.

Indyk, P., Koudas, N., and Muthukrishnan, S. 2000. Identifying representative trends in massive time series data sets using sketches. In Proc. of the 26th International Conference on Very Large Data Bases, Cairo, Egypt, pp. 363–372.

Jain, A.K., Murty, M.N., and Flynn, P.J. 1999. Data clustering: A review. ACM Computing Surveys, 31(3):265–323.

Kalpakis, K., Gada, D., and Puttagunta, V. 2001. Distance measures for effective clustering of arima time-series. In Proc. of the IEEE International Conference on Data Mining, San Jose, CA, pp. 273–280.

Keogh, E. and Smyth, P. 1997. A probabilistic approach to fast pattern matching in time series databases. In Proc. of the 3rd International Conference on Knowledge Discovery and Data Mining, Newport Beach, CA, USA, pp. 20–24.

Keogh, E., Chakrabarti, K., Pazzani, M.J., and Mehrotra, S. 2001. Locally adaptive dimensionality reduction for indexing large time series databases. In Proc. of the ACM SIGMOD Conference on Management of Data, Santa Barbara, CA, USA, pp. 151–162.

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

Keogh, E. and Kasetty, S. 2002. On the need for time series data mining benchmarks: A survey and empirical demonstration. In Proc. of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, Alberta, Canada, pp. 102–111.

Keogh, E., Lin, J., and Truppel, W. 2003. Clustering of time series subsequences is meaningless: Implications for past and future research. In Proc. of the 3rd IEEE International Conference on Data Mining, Melbourne, FL, USA, pp. 115–122.

Keogh, E., Lonardi, S., and Ratanamahatana, C. 2004. Towards parameter-free data mining. In Proc. of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, WA, pp. 206–215.

Kohonen, T., Oja, M., Kaski, S., and Somervuo, P. 2002. Self–Organizing map. Biennial report 2000–2001.

Lange, T., Roth, V., Braun, M.L., and Buhmann, J.M. 2004. Stability-based validation of clustering solutions. Neural Computation, 16(6):1299–1323.

Lee, T.-H. 2001. Neural network test and nonparametric kernel test for neglected nonlinearity in regression models. Studies in Nonlinear Dynamics & Econometrics, 4(4):169–182.

Lin, J., Vlachos, M., Keogh, E., and Gunopulos, D. 2004. Iterative incremental clustering of time series. In Proc. of the IX Conference on Extending Database Technology, Crete, Greece, pp. 106–122.

Lu, Z.-Q. 1996. Estimating lyapunov exponents in chaotic time series with locally weighted regression, Ph.D. Thesis, Department of Statistics, University of North Carolina.

Makridakis, S., Wheelwright, S.C., and Hyndman, R.J. 1998. Forecasting methods and applications. John Wiley & Sons, Inc.

Mörchen, F. 2003. Time series feature extraction for data mining using dwt and dft. Technical Report No. 33.

Nanopoulos, A., Alcock, R., and Manolopoulos, Y. 2001. Feature-based Classification of Time-Series Data. International Journal of Computer Research. NY: Nova Science Publishers, pp. 49–61.

Popivanov, I. and Miller, R.J. 2002. Similarity search over time series data using wavelets. In Proc. of the 18th International Conference on Data Engineering, San Jose, CA, USA, pp. 212–221.

Pyle, D. 1999. Data Preparation for Data Mining. San Francisco, California: Morgan Kaufmann Publishers, Inc.

Ratanamahatana, C.A. and Keogh, E. 2005. Three myths about dynamic time warping. In Proc. of the SIAM International Conference on Data Mining, Newport Beach, CA, pp. 506–510.

Rocca, M.L. and Perna, C. 2004. Subsampling model selection in neural networks for nonlinear time series analysis. In Proc. of the 36th Symposium on the Interface, Baltimore, Maryland,

Rose, O. 1996. Estimation of the Hurst Parameter of Long-Range Dependent Time Series. Research Report, 137.

Royston, P. 1982. An extension of shapiro and wilk's w test for normality to large samples. Applied Statistics, 31:115–124.

Scargle, J.D. 2000. Timing: New methods for astronomical time series analysis. Bulletin of the American Astronomical Society, 32:1438.

Teräesvirta, T, Lin, C.F, and Granger, C.W.J. 1993. Power of the neural network linearity test. Journal of Time Series Analysis, 14(209–220)

Teräesvirta, T. 1996. Power properties of linearity tests for time series. Studies in Nonlinear Dynamics & Econometrics, 1(1):3–10.

Van Laerhoven, K. 2001. Combining the knohonen self-organizing map and k-means for on-line classification of sensor data. Artificial neural networks, lecture notes in artificial intelligence. Springer Verlag, pp. 464–70.

Wallace, C.S. 1999. Minimum Description Length. The Mit Encyclopedia of the Cognitive Science. The MIT Press, London, England, pp. 550–551.

Wang, C. and Wang, X.S. 2000. Supporting content-based searches on time series via approximation. In Proc. of the 12th International Conference on Scientific and Statistical Database Management, Berlin, Germany, pp. 69–81.

Willinger, W, Paxon, V, and Taqqu, M.S. 1996. Self-similarity and heavy tails: Structural modeling of network traffic. A Practical Guide to Heavy Tails: Statistical Techniques and Applications: 27–53.

Wolf, A. Swift, J.B. Swinney, H.L. and Vastano, J.A. 1985. Determining lyapunov exponents from a time series. PHYSICA D, 16:285–317.

Wood, S.N. 2000. Modelling and smoothing parameter estimation with multiple quadratic penalties. J. R. Statist. Soc. B, 62(2):413–428.