Efficient and Scalable Techniques for Multivariate Time Series Analysis and Search

Aminata Kane

A Thesis
In the Department
of
Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of
Doctor of Philosophy (Computer Science) at
Concordia University
Montréal, Québec, Canada

August 2017

Concordia University
School of Graduate Studies

This is to certify that the thesis prepared

By: **Ms. Aminata Kane**

Entitled: **Efficient and Scalable Techniques for Multivariate Time Series Analysis and Search**

and submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy (Computer Science)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining commitee:

_____ Chair
Dr. Luis Amador

_____ External Examiner
Dr. Davood Rafiei

_____ External to Program
Dr. Jamal Bentahar

_____ Examiner
Dr. Adam Krzyzak

_____ Examiner
Dr. Todd Eavis

_____ Thesis Supervisor
Dr. Nematollaah Shiri V.

Approved _____
Dr. Volker Haarslev, Graduate Program Director

August 21, 2017

_____
Dr. Amir Asif, Dean
Faculty of Engineering and Computer Science

Abstract

Efficient and Scalable Techniques for Multivariate Time Series Analysis and Search

Aminata Kane, Ph.D.
Concordia University, 2017

Innovation and advances in technology have led to the growth of time series data at a phenomenal rate in many applications. Query processing and the analysis of time series data have been studied and, numerous solutions have been proposed. In this research, we focus on multivariate time series (MTS) and devise techniques for high dimensional and voluminous MTS data. The success of such solution techniques relies on effective dimensionality reduction in a preprocessing step. Feature selection has often been used as a dimensionality reduction technique. It helps identify a subset of features that capture most characteristics from the data. We propose a more effective feature subset selection technique, termed Weighted Scores (WS), based on statistics drawn from the Principal Component Analysis (PCA) of the input MTS data matrix. The technique allows reducing the dimensionality of the data, while retaining and ranking its most influential features. We then consider feature grouping and develop a technique termed FRG (Feature Ranking and Grouping) to improve the effectiveness of our technique in sparse vector frameworks. We also developed a PCA based MTS representation technique M2U (Multivariate to Univariate transformation) which allows to transform the MTS with large number of variables to a univariate signal prior to performing downstream pattern recognition tasks such as seeking correlations within the set. In related research, we study the similarity search problem for MTS, and developed a novel correlation based method for standard MTS, ESTMSS (Efficient and Scalable Technique for MTS Similarity Search). For this, we uses randomized dimensionality reduction, and a threshold based correlation computation. The results of our numerous experiments on real benchmark data indicate the

effectiveness of our methods. The technique improves computation time by at least an order of magnitude compared to other techniques, and affords a large reduction in memory requirement while providing comparable accuracy and precision results in large scale frameworks.

Dedication

To my family.

Acknowledgments

This dissertation would not have materialized without the support of many, to whom I owe much appreciation and gratitude.

My deepest gratitude to my supervisor Dr. Nematollaah Shiri for providing the platform to complete this work, for his valuable advices, and encouragements throughout the course of this thesis. His boundless patience and availability have been instrumental in helping me see this dissertation through.

I also wish to express my sincere gratitude to Dr. Jamal Bentahar, Dr. Todd Eavis and Dr. Adam Krzyzak, for serving on my doctoral research committee, for their availability and interest in the work, for their valuable comments and feedbacks at different stages of this thesis.

I am much indebted to all members of Concordia University who from near or far contributed in making the completion of this this work possible at Concordia University. I particularly extend my sincere thanks to Halina Monkiewicz for her availability, patience and encouragements throughout my thesis, to Dr. Volker Haarslev for his patience and valuable comments during my proposal defense.

I gratefully acknowledge two of my professors from the University of Massachusetts Dr. Alexander Olsen and Prof. William Moloney whose support have been instrumental in allowing me to pursue doctoral research in my chosen field of study.

I extend special thanks to my friends from Torc Financial LLC: Vladimir Panasenko, Kshama Tanga, Raja Sundararaman, Dr. Robert Bergelson, Satish Jeyaram, who sparked in me the interest for big data analytics and my friends at Concordia University: Mahsa Orang, Laleh Roostapour, Ali Moallemi, Shayan Manoochehri, Shahab Harrafi, Iraj Hedayati, who made my stay at Concordia more enjoyable.

Finally, words cannot express how grateful and appreciative I am of my family for their unconditional love and support, their enduring patience, their availability at any time of the day or night, their guidance and motivations, and for their efforts to provide me with the best possible education.

This research was supported in part by Natural Sciences and Engineering Research Council (NSERC) of Canada and Concordia University.

Table of Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1: Introduction

1.1 Time Series

A time series consists of observations recorded on discrete time points (discrete time series) or continuously through time (continuous time series) at regular time interval. Continuous time series can be discretized without much loss of information by using its inherently discrete type or techniques such as sampling or aggregation. Time series can be univariate or multivariate in nature.

$$\begin{pmatrix} -0.80 \\ -0.29 \\ -0.26 \\ -0.45 \end{pmatrix} \qquad \begin{pmatrix} -0.80 & -0.51 & -1.36 \\ -0.29 & -0.51 & -1.43 \\ -0.26 & -0.70 & -1.07 \\ -0.45 & -1.02 & -0.73 \end{pmatrix}$$

(a) (b)

Table 1.1: Vector representation of a UTS (a), Matrix representation of a MTS (b)

	t1	t2	t3	t4
Var1	-0.80	-0.29	-0.26	-0.45

(a)

	t1	t2	t3	t4
Var1	-0.80	-0.29	-0.26	-0.45
Var2	-0.51	-0.51	-0.70	-1.02
Var3	-1.36	-1.43	-1.07	-0.73

(b)

Table 1.2: Table representations of a UTS(a) and MTS(b)

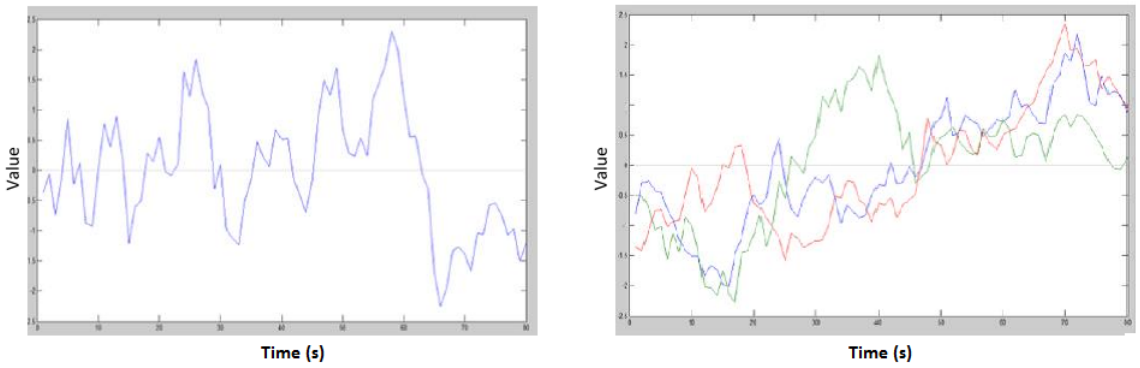

Figure 1.1: Signal(s) representating a UTS(Left) and UTS(right)

A univariate time series (UTS) T of length n pertains to one variable. It can be viewed as a point in an n dimensional space and expressed as: T= <x_1, x_2, ..., x_n>, where x_i is a real value in $\mathbb{R}$.

A multivariate time series (MTS) refers to time series that deal with recordings of values for more than one variable/attribute at regular interval of times. It can be viewed as a number of UTS. If m is the number of variables in a MTS A, we can write: $A = (a_{11}, \ldots..a_{1n}), (a_{21}, \ldots a_{2n}), \ldots. (a_{m1}, \ldots.a_{mn})$.

Then, a MTS A that has n instances (of length n) and m variables can be represented as an n × m element matrix $A_{n\times m}$, with n rows and m columns a follows:

$$A_{n,m} = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,m} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,m} \end{bmatrix}$$

Time series data represent a large fraction of the world's supply of data [97] and, data generated in many applications can be transformed into time series without much loss of information [58, 56]. Hence, a growing number of applications in areas such as Finance, Neuroscience, health sciences require the ability to analyze and process such voluminous data. Unfortunately, when it comes to processing large time series datasets, the challenges already pertaining to time series in reduced settings (e.g. high dimensionality, redundancy or noise introduced through data collection

and the presence of dependencies between features) are of greater scale. On the other hand, most classical techniques are not adequately equipped to gracefully scale to larger datasets. Hence, pattern recognition on such data involves either tweaking the existing techniques or coming up with new ones that would adaptively process data in such environment.

Research in this field has been particularly active in recent years. While much has been achieved, the proposed techniques mostly focus on UTS and are not easily applicable to real life scenarios known to be better captured in multivariate abstractions. Examples include the global economy, in which coexist a number of markets around the globe, trading a variety of financial products daily. Although one could be brought to think that, geographical locations or country local sovereignties and monetary regulation would make of the global financial markets a group of independent structures, it has been shown that asset prices often respond to the same global events [2]. In Neuroscience, the complexity of the human brain; its need for strong inter-connectivity and cohesion is better captured by also analyzing how voxels or regions of interest relate to one another than by merely analyzing voxels individually [85, 34, 52]. A multivariate analysis of time series data often considers the phenomena from an overall perspective, determines and leverages the intrinsic structure of the multivariate data, while providing a multi-layered analysis for better insights. In particular, research using correlation based techniques [48, 51, 100, 103] such as the Canonical Correlation Analysis(CCA) or Principal Component Analysis (PCA) have shown that capturing and leveraging the information within MTS can be crucial in improving efficiency in many data mining areas such as similarity search, feature-subset selection, clustering, classification, hence the importance of developing techniques that would work better for multivariate series with large number of variables.

1.1.1 Notation

This section provides the notation used in this thesis unless otherwise specified.

- D denotes the set of MTS (or, D' if normalized).

- D^U denotes the set of UTS.
- $A_{n,m}$ denotes a multivariate TS of n instances with m variables.
- A, A $= [a_{ij}]$ denotes a matrix representing the multivariate TS.
- A^T denotes the matrix transpose of A.
- V is the right eigenvector matrix of size $m \times m$, V_k is the right eigenvector matrix of size $m \times k$.
- S is the diagonal matrix of the singular values of A, S_k is the diagonal matrix of the k largest singular values of A.
- $a_{i,*}$ denotes the i^{th} row of the matrix.
- $a_{*,j}$ denotes the j^{th} column of the matrix.
- $a_{i,j}$ denotes the element entry at the i^{th} row and j^{th} column of the matrix.
- θ the explained variance in the data that are represented within k retained principal components
- ρ is the Pearson correlation coefficient.
- ϵ is the user specified correlation threshold.

1.2 Thesis Objectives

The foundation of this research stems from two important aspects. First of which is the need to devise techniques that are better suited for today's big data characteristics. And, the second aspect is the necessity to analyses and process data from highly unified frameworks such as an enterprise risk, the human brain, or a human organisms as multivariate concepts and rules rather than merely a concurrence of univariate ones. Highly unified frameworks present multilayed complexities that are better captured and conveyed through multivariate studies.

This research is focused on devising efficient and scalable techniques for time series data analysis and search, particularly for MTS, where we primarily relied on the PCA.

The presented techniques retain the crucial underlying characteristics of the original MTS data, and leverage its structural properties for better interpretation. They size-ably reduce the time complexity and memory requirements for large datasets where storing the whole data in memory or relying on a large time complexity is not an option.

1.3 Thesis Contributions

This thesis presents a number of efficient and scalable techniques for MTS analysis and search. Our contributions, described as follows, particularly reside in the domains of dimensionality reduction and similarity search.

- *A feature subset selection technique, the Weighted Scores(WS) technique [48]*: We first study the problem of uncovering the most relevant and discriminative features in a MTS. We analyze the MTS internal structure to find and leverage more information about the variables as they all do not equally contribute to the MTS. The technique relies on statistics drawn from the PCA to determine the weights of the variables with respect to the whole multivariate dataset and rank them accordingly. Subsequently selecting the set of most relevant features allows reducing the dimensionality while retaining the domain interpretability. The technique is unsupervised and sets a framework for improved efficiency in time series pattern recognition tasks such as classification, clustering or similarity search.

- *A feature subset selection and grouping technique, the FRG (Feature Ranking and Grouping technique) [49]*: In some practical applications, feature subset selection alone may disregard important information when seeking for the most relevant and discriminative features in MTS. Those frameworks include MTS data exhibiting sparse feature vector structures, or Bio-informatics applications where dependent features are known to work better in groups than on their

own. In such cases combining feature selection to feature grouping yields better results [110]. We present an unsupervised feature selection and grouping technique, namely FRG, that reduces noise, identifies relevant features, and groups correlated ones for increased efficiency and accuracy. The technique uses unsupervised learning through randomized PCA to determine influence and rank the features accordingly. Correlated features are then subsequently identified, grouped, and recombined into unique features to allow for a more efficient and scalable processing of high dimensional MTS.

- *A MTS reduction and representation technique, termed M2U(Multivariate to Univariate transformation) [50]*: We present a PCA based MTS transformation technique that converts the MTS with large number of variables to a UTS prior to performing downstream pattern recognition tasks such as seeking correlations within a set of UTS. This technique is particularly important because, on one hand, the transformation takes into account the correlation between variables, in addition to decreasing redundancy and noise and, reducing the intrinsic high dimensionality. Other proposed univariate representations are often not able to retain the correlation between variables within each multivariate dataset. On the other hand, substantial recent research studied ways to improve efficiency for UTS pattern recognition tasks in general, and similarity search in particular [79, 88, 12, 71]. Our proposed representation will allow efficient UTS techniques to be easily extended to MTS data.

- *A UTS transformation and representation technique, TVR(Trend and Value Representation)* [47]: We developed a UTS transformation and representation technique, TVR(Trend and Value Representation). This technique is obtained by extending the clipping technique [81, 53, 54] and incorporates the time series trend information, in addition to the value information in order to better capture the time series characteristics and providing greater accuracy.

- *The formulation of a similarity measure [47] based on a binary weighted dissimilarity measures for mixed types of variables measuring different objects*: We

formulate of a weighted symbolic similarity measure measure based on a binary weighted dissimilarity measures for mixed types of variables measuring different objects. Using this similarity measure along with TVR in the pruning phase allows to substantially reduce the search space in large dataset frameworks.

- *An efficient and scalable technique for MTS Similarity Search:* We proposed the use of the three techniques M2U [50], TVR [47] and the proposed symbolic correlation measure based on a binary weighted dissimilarity measure for mixed variables [47] in conjunction, to devise an efficient and scalable technique for MTS similarity search (ESTMSS). The technique improves computation time by at least an order of magnitude compared to other techniques, and affords a massive reduction in memory requirement while providing comparable accuracy and precision results in large scale frameworks.

These techniques contribute, from a general perspective, to the effort looking to address two sizable challenges that this area encounters: the high dimensionality of the data which makes it difficult to work with; and the scarcity of effective similarity search techniques for MTS.

1.4 Thesis Organization

The remainder of this dissertation is organized as follows. We review the background and related literature in Chapter 2. Chapter 3 presents the feature subset selection technique Weighted Scores (WS), followed by the feature subset selection and grouping technique FRG in Chapter 4. The MTS reduction and representation technique M2U is introduced in Chapter 5. Chapter 6 discusses the UTS transformation and representation technique, TVR (Trend and Value Representation), followed by the formulation of a similarity measure based on a binary weighted dissimilarity measures for mixed types of variables measuring different objects. Concluding remarks and future directions are presented in Chapter 7.

Chapter 2: Background and Related Work

Multivariate time series (MTS) data mining presents major challenges and, a fair amount of pre-processing is often required to improve the usability of the data for downstream pattern recognition tasks such as similarity search. The greatest challenges encountered stem from the high dimensionality of the data, both in terms of length and number of variables, its volume and, the need to accurately assess similarities in time series. Core research activities in this area can essentially be classified into the following three areas: (1) time series data reduction and transformation, (2) time series similarity measures, and (3) time series indexing. Reductions and transformation techniques often look to uncover a reduced representation while retaining the important characteristics of the original data. Similarity measures help identify patterns and shapes. They assess how alike or different are time series based on given criteria. Time series indexing structures and techniques support efficient computation in terms of time and memory requirements. Our research goals and contributions fit in the first two core areas. In what follows we introduce the background and literature pertaining to those two core areas.

2.1 Time Series Data Reduction Techniques

Data reduction is widely recognized as an important preprocessing step for pattern recognition tasks; especially in large data frameworks. In the particular case of time

series, data reduction can often be seamless. This is due to the fact that time series data inherently presents a structure such that, it generally exhibits some amount of redundancy. A given point may influence many nearby observations though auto-correlation, and two successive data points can often be within a predictable range, hence presents the possibility to seamlessly reduce the data.

While many data reduction and representation strategies have been proposed, the techniques must be carefully chosen to ensure their suitability for the data at hand and, for the intended downstream tasks. Doing so, ensures the effectiveness of the overall mining technique. Time series data reduction techniques can be generally categorized in three core areas: data compression, numerosity reduction, and dimensionality reduction; although dimensionality reduction and numerosity reduction may be considered as forms of data compression.

Data compression techniques primarily provide a strategy to minimize the amount of data (or number of bits) needed to be stored or transmitted. Compressions can be qualified as either lossless or lossy. Lossless compressions generally rely on data redundancies to reduce the data without losing information, and hence, allow for a recovery of the full information when uncompressed. Lossy compressions on the other hand, rely on strategies that reduce the data by omitting nonessential information according to human perception for instance. They ideally present an acceptable level of information loss with respect to the gain in other aspects such as memory space or time complexity.

Numerosity reduction techniques use parametric and non-parametric models such as sampling, histograms or clustering to estimate the original data and replace it by smaller forms of representation. For instance, in the case of the parametric model, once the data has been estimated using the parametric model, storing the parameter rather the original data is common practice.

Dimensionality reduction techniques seek to uncover or devise a rich reduced set of features that retain crucial underlying structure from the original data. More formally stated: given a set of n points $X = \left\{n \in \mathbb{R}^d\right\}$ in a high dimensional Euclidien space, we look to describe X in fewer dimensions $k << n, d$ without distorting the Euclidien distance between any two points by much (by less than a well defined small

ϵ).

Otherwise expressed, define a function f such that $\forall x_i, x_j \in X$

$$\|f(x_i) - f(x_j)\|_2 \approx \|x_i - x_j\|_2. \tag{2.1}$$

Dimensionality reduction techniques include feature extraction, feature selection or feature re-engineering techniques. The state or art techniques are often based on one or a combination of some of four widely used techniques: (1)Fourier transform, or a derivative of the Fourier transform (2) the Wavelet transform, (3)the Singular Value Decomposition, and the (4)Random projection technique. Depending on the type of data at hand, some techniques may be more suitable to consider than others. In what follow, we introduce those four widely used techniques but first define some notions that they shared.

Definition 2.1. *(Orthogonal Transformation) Given two vectors $\vec{u}$ and $\vec{v}$, an orthogonal transformation T: V -> V of $\vec{u}$ and $\vec{v}$ to T($\vec{u}$) and T($\vec{v}$) respectively is a linear transformation which preserves symmetric inner product. In particular, it preserves the length of the vectors and the angle between the vectors.*

Definition 2.2. *(Orthonormality) - Given two vectors $\vec{u}$ and $\vec{v}$ in an inner product space (a vector space that has an additional structure called inner product, generalizing the dot product) , $\vec{u}$ and $\vec{v}$ are said to be orthonormal if they are both unit vectors (their lengths are each equal to 1) and are orthogonal (the angle between them is 90). An orthonormal set of vectors is comprised of vectors that are all pairwise orthogonal and of unit length.*

The **Fourier transform** [23], originated from results first introduced by Jean Baptist Joseph Fourier in 1807 stating the feasibility of expressing any piecewise continuous periodic function with period T, as the sum of a (possibly infinite) set of oscillating functions, based on the sines and cosines and, whose frequencies are multiple of the angular frequency $\omega_0 = \frac{2\pi}{T}$. Those results were further extended to any periodic function f(t) with period T. Such functions may hence be approximated to

a sum of their Fourier series (or complex exponentials) and expressed as:

$$f(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} (a_k cos\ \frac{2k\pi t}{T} + b_k sin\ \frac{2k\pi t}{T}) \qquad \text{where:}$$

$$a_0 = \frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(t)dt,$$
$$a_k = \frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(t)cos\frac{2k\pi t}{T}dt, \qquad \text{k = 1, 2, 3, ...}$$
$$b_k = \frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(t)sin\frac{2k\pi t}{T}dt, \qquad \text{k = 1, 2, 3, ...}$$

The Fourier transform decomposes and analyses a function of time into its frequency domain, rather than its time domain. For data that is discrete in nature such as time series, the Discrete Fourier Transform (DFT) is more appropriate. This technique is based on orthogonal transforms where the coefficients are uncovered by carrying out an inner product between the input signal and a set of orthonormal basis functions. It is often preferred when the data is periodic due to the periodic nature of the functions used to approximate the original data. The original time series data is transformed into frequency components sorted in decreasing order of importance. As the first few components often carry most of its so called energy, ignoring the remaining components known as negligible is appropriate. Once the dimensionality is reduced, the Fourier domain affords a framework where data processing is more efficient. An approximation of the original features can be reconstructed by using the available components.

The time complexity of the fast Fourier transform is such as for a time series of length n, computing the k first component takes Min (O(nlogn), O(kn)). Clearly, although acceptable some other techniques such as the Discrete Wavelet transform (DWT) reviewed next, present a better time complexity.

The **Wavelet transform** is an orthogonal transform and a derivative of the Fourier transform. The first wavelet transform known as Haar wavelet [28] was introduced by Alfred Haar in 1909 although wavelets were not yet defined as such until Jean Morelet coined the concept in 1981. Similarly to the way the Fourier transform relies on the *sin* and *cos* functions as its basis functions to decompose other functions,

the wavelet transform leverages *wavelets* as its basis functions. With $\Psi(t) \in L^2(R)$ set as the mother wavelet, it may be transformed and dilated through critical sampling. The wavelet transform may be expressed as:
$Wf(a,b) = \int_{-\infty}^{\infty} f(t) \frac{1}{\sqrt{a}} \Psi^{\star} \left(\frac{t-b}{a}\right) dt \mid_{b \in R, a \in R^{+}}$
where b is the translating parameter, indicating the corresponding region, a is the scaling parameter greater than zero because negative scaling is undefined.

In this framework as well, the discrete wavelet transform (DWT) is more appropriate for data with discrete time points such as time series and an inner product between the input signal and a set of orthonormal basis functions allow to uncover the coefficients.

The family of DWT present a key advantage over the DFT in that they are multi-resolution transforms, hence allow to efficiently operate in both the spectral and time domains. Indeed, although a particular case of the Fourier transform known as the Short-time Fourier transform (STFT) may provide both time and frequency information, the resolution in frequency may be limited by the fixed size sliding window used to uncover its spectrogram. The Wavelet transform affords a framework that not only enables the intrinsic analysis of the time series' frequency but also provides other important insights such as the scale of the time series or time at which a specific observation occurred.

It is also possible to reconstruct an approximation to the original time series by using the coefficients that result from the DWT. The scale of the transformation, or amount of details in the signal, plays an important role in the reconstruction process. Coarser resolution coefficients pertain to small scales and allow for better data reconstruction than the higher resolution coefficients that refer to larger scales. The time complexity of the DWT for a signal of length n is O(n) (wavelet transforms without compact support may however require $O(n^2)$).

The **Singular Value Decomposition** (SVD) is an orthogonal linear transformations in which one assumes all basis vectors to form an orthonormal matrix. It projects the original dataset in a new coordinate system where the directions are pairwise orthonormal. It has been used in many applications primarily for the following key advantages:

- reduces redundancies and noise introduced during the data collection process.
- reduces the number of variables while retaining the variability in the data.
- identifies relationships and interactions between variables.
- identifies hidden patterns and classify them according to the amount of information stored in the data.

The SVD may be used as an important step in many other powerful dimensionality reduction techniques such as Principal Component Analysis(PCA) [48], Canonical Correlation Analysis(CCA) [9] or Independant Component Analysis(ICA) [33] among others. The basis vectors of the SVD are however data dependent which presents both advantages and disadvantages. Among the disadvantages is the need to store the basis vectors in addition to the new data points in order to be able to reconstruct the original data [114]. Although this technique currently provides the best approximation to an original matrix, computing the SVD of a large matrix of n instances and m variabes can be expensive with a time complexity of O(min(n^2m, m^2n)) if randomization techniques are not used.

The **Random Projections technique**, unlike the DFT, DWT, SVD, is not based on orthogonal transformations, but rather on a projection of time series randomly to a lower dimensional space. It is fundamentally based on the Johnson-Lindenstrauss lemma proposed in 1984 [40] and stated as:

Lemma 2.1. *(Johnson-Lindenstrauss Lemma)*
For any $0 < \epsilon < 1$ *and any interger* n *let* k *be a possitive interger such that*

$$k \geq \frac{24}{3\epsilon^2 - 2\epsilon^3} \log n \tag{2.2}$$

then for any set A *of* n *points* $\in \mathbb{R}^d$
there exists a map $f : \mathbb{R}^d \to \mathbb{R}^k$ *such that for all* $x_i, x_j \in A$

$$(1-\epsilon)\|x_i - x_j\|^2 \leq \|f(x_i) - f(x_j)\|^2 \leq (1+\epsilon)\|x_i - x_j\|^2 \tag{2.3}$$

The lemma conveys that, given a set of points in a high-dimensional space, they can be projected and embedded into a lower dimensional subspace, such that distances between the points are nearly preserved.

For random projections, the lower dimensional subspace is randomly chosen based on some distribution and, we can seek to have a probabilistic guaranty that the distance between two time series in the higher dimensional space will have some sort of correspondence with the distance between the same two time series in the lower dimensional space. Considering a matrix $A_{n\times m}$ the original data with m variables and n observations, then $A_{k\times m} = R_{k\times n}A_{n\times m}$ is the random projection of $A_{n\times m}$ onto a lower k-dimensional subspace. This technique is carried out by using a random matrix whose rows have unit lengths $R_{k\times n}$ and projecting the original n-dimensional data onto a k-dimensional ($k << n$) subspace. The time complexity of such a transformation is O(nkm). Unlike DFT, DWT and SVD, the random projection method does not allow to reconstruct the original data. This technique is known to be efficient for frameworks with relatively small numbers of very long time series due to the fact that, the data size k resulting from the reduction does not depend on the time series instances but on the number of time series [114]. It is however known to be less effective than PCA for severe dimensionality reduction [24].

In this thesis work, since our intent is to substantially reduce the dimensionality without impacting time and accuracy, we essentially use either the standard SVD, or randomized versions of the SVD [30, 17] in our computation of the PCA (discussed in Chapter 3), which allows to achieve those objectives as we will see in Chapters 3, 4, 5 and 6.

Dimensionality reduction techniques often provide time series representations strategies that afford a framework where large scale data becomes more manageable. Such representation techniques can primarily be grouped into four classes [81], as illustrated in Fig. 2.1. The classes are labeled according to the nature of transformation applied to the data: model based, data adaptive based, non-data adaptive, and data dictated. *Model based time series representation techniques* often use the parameters driving the model to represent the time series. In this case, comparing two times series may be carried out by checking whether they have the same set of parameters driving the

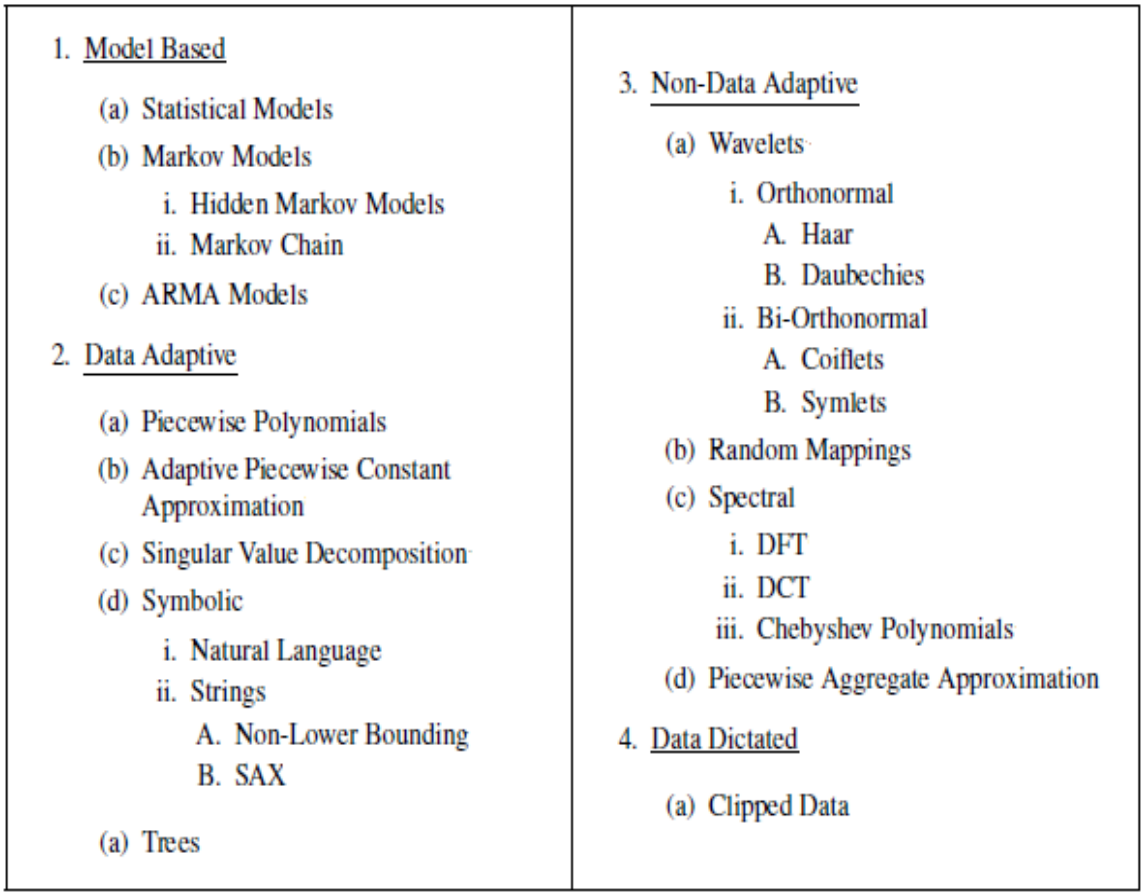

Figure 2.1: Time series representation techniques partly extracted from the taxonomy in [81]

model. *Non-data adaptive techniques* rely on the same set of transformation parameters for all types of data, regardless of their specificities or of the differences between their respective features. Most non-data adaptive techniques can be transformed to data adaptive techniques by adding a data sensitive selection step [21]. *Data adaptive representation techniques* assume that, during the transformation process, parameters of the transformation such as the compression ratio of the data are influenced by the internal structure of the data. The SVD for instance is a data adaptive representation technique that works particularly well for data where linear dependencies exist. *Data dictated representation techniques* rely on a compression ratio that is data dictated.

For instance, in the particular case of of the clipping technique, the time series recordings are transformed into bits, where each bit value is indicative of the corresponding point's position with respect to the average. Clipped data [81, 53, 54], consisting of a sequence of symbolic bits has the advantage of being directly comparable to the raw time series data.

Our proposed Trend and Value Representation(TVR) technique incorporates techniques from three different classes: a non-data adaptive technique(Piecewise Aggregation Approximation(PAA)), a data adaptive technique (Symbolic Aggregation Approximation(SAX)) and a data dictated technique (clipping); although clipping can be considered a particular case of the SAX representation [81]. Hence, this allowed to leverage advantages from all three classes of techniques as we will see in Chapter 6.

2.2 Multivariate Time Series Similarity Search Related Literature

Similarity search can either be regarded as a stand-alone task or as a crucial step in many data mining tasks such as indexing, classification, clustering or anomaly detection [79].

An important step in a similarity search process is devising a similarity measure to assess how alike or different are time series based on given criteria. When seeking similarities between time series, one can consider many aspects, among which similarity in time, in shape or in change. The similarity in time otherwise implies how correlated are time series. The similarity in shape seeks for the likeness in patterns of change, irrespective of the time points. The similarity in change considers the likeness in auto-correlation structure.

The choice of the similarity measure often depends on a few aspects among which the type of data at hand, the feature(s) that we look to compare, and the type of application we are dealing with.

Similarity measures can primarily be classified in to four categories based on the

paradigms that guide them: shape based measures, edit based measures, feature based measures, and structure based measures. Shape based measures seek to compare the general shape of two time series. Edit based measures tend to first transform the time series to a alternate representation then to assess the minimum number of operations required for the transform to occur. Feature based measures investigate and extract important features from original time series then, subsequently compare the extracted features using a distance measure. The aim of the structure based measures is a bit similar to that of the shape based measures, with respect to the fact that they look to compare the time series from an overall perspective. The structure based measures often seek to find high level structures in the data, which will allow for a comparison in an overall perspective. The structure based measures are further considered compression based when they analyses how well the time series can be compressed together; higher compression ratios make better similarity measures. On the other hand they are considered model based when the strategy is to fit a number of time series to a model, then assess the similarity by comparing the model's parameters.

The work in this thesis investigated similarities in time and shapes, while linear dependency was an important factor. We essentially used Pearson correlation as the measure to assess similarity between two time series.

Research in time series similarity search has attracted the attention of many researchers, resulting in much progress in recent years, particularly for UTS. We can divide the research activities and results since the 90^s to three periods.

The early years, from 1994 to 2003 represent a period which described concepts along with the technique strategy and provided implementation guidelines. The proposed techniques in this time period mainly focused on UTS and smaller dataset sizes.

During 2003 to 2008, two tendencies were more pronounced. A first one where the research investigated techniques proposed at the time and looked to improve them, or devise more efficient strategies. A second tendency reflected a moment where the research community had started to understand the importance of devising techniques for MTS. Indeed, it became clear that many situations are in fact only part

of complex occurrences that involve many interrelated aspects. Such situations are better represented by multivariate concepts and rules, than by compounded isolated effects from univariate ones. A MTS data analysis affords a multi-layered analysis, uncovers interactions between variables, and provides an overarching view of the internal structure for better insights. Although there were serious attempts at the time and good progress was achieved, the literature often focused on MTS with low number of variables.

Since 2008, several solution techniques were proposed to improve similarity search and related applications such as clustering or classification. We can note three particular shifts in the current literature.

The first shift represents a tendency of moving away from theoretical "Toy problems to realistic deployments" [35] to address the real world problems. For instance, a clear effort to develop techniques that would process time series data as received, with its imperfections (uncleansed data, different length series, etc.) [35, 36, 92].

The other two shifts are found in the literature looking to address today's big data challenges. Those challenges essentially stem from its magnified characteristics in terms of volume, variety, increased need for velocity, veracity and value.

Where one focuses on processing scalable and efficient search in large size data, the second one focuses on scalable and efficient search in MTS with larger numbers of variables and dimensions. The latter is the focus of this thesis work.

The research trend focused on efficient and scalable search in large size data is mostly directed to UTS (search across large numbers of UTS, millions to trillions of them) and heavily relies on indexing. In particular, we note the work of Shieh and Keogh [87] who proposed a technique for indexing and mining terabyte sized time series, or Camerra et al. [12], for indexing and mining a billion of time series in later work. In 2012, a study by Rakthanmanon et al. [79] proposed a method for searching and mining trillions of time series subsequences by combining 4 techniques, allowing for early abandoning and using Dynamic Time Warping. "The difficulty of scaling search to large datasets largely explains why most academic work on time series data mining has plateaued at considering a few millions of time series objects, while much of industry and science sits on billions of time series objects waiting to be

explored" [79]. While much has been achieved with the techniques in this trend, they mostly focus on indexing techniques and on UTS. More work is required to develop techniques that would work better for MTS.

The second research trend focuses on efficient and scalable search for MTS with a large number of variables (and often higher number of instances) that generally follows one of three approaches. In the first approach, each variable within the MTS is considered independently as a time series [22], and often analyzed separately by using univariate techniques. While being easier to process, this approach often requires much more computation time. The second approach consists of stacking all data contained within all variables as a lengthy UTS [46], to be analyzed as such, using univariate techniques. Like the first approach, this strategy often overlooks the relationships that exist among the variables and cannot efficiently process a relatively large number of variables. The third approach, considers the MTS as a whole and transforms it into a lower dimensional representation that still captures its main characteristics, and the hidden relationships among variables, while rendering the data more manageable [86, 103, 82]. Although this approach presents more complexity, it provides more accurate results for similarity search. Techniques of this third approach can often be classified into two groups; those that deal with streaming time series data, and the rest. In the following subsections we will review these techniques.

2.2.1 Multivariate Time Series Similarity Search Techniques for Data in Motion (Streaming Data)

The first group of techniques, proposed to identify patterns in streaming time series, which is continuous, unbounded, and a timely ordered sequence of data elements generated and/or collected (at a rapid rate). These series may include multiple measurements at each time point hence, they can be looked at as MTS.

A notable work in this area is an approach for discovering correlations and hidden variables in multiple streaming time series, introduced through SPIRIT((Streaming Pattern dIscoveRy in multIple Timeseries) by Papadimitriou et al. in [75]. The discovered information would be used to summarize the entire trend in the set of streams

and, allow for fast forecasting and outliers detection. This technique is based on PCA and satisfies some important goals of online streaming algorithms such as automatic detection of patterns, adapting to the changes in real-time, scaling up linearly with the number of time series and, suitability for distributed environments.

In a practical application, Sayal et al. [84] introduced a method for extracting time correlations among multiple stream time series. The proposed method expresses time correlations as reusable correlation rules and textual representations conveying observed dependencies or relationships among stream time series. For instance, the observation that a change in the values of one set of time-series data streams initiates a chain effect that causes the change in the values of another set of time-series data streams. For example, if the value of an attribute A increases more than 5%, then the value of the other attribute B is expected to decrease by more than 10% within two days. More precisely, in an IT operational environment, a server performance has a time-delayed impact on database performance that can be quantified according to proposed rules. The proposed method consists of three main steps. In the first step, the original time series data was summarized by aggregation at different time granularities. In the second step, the change points upon which the comparisons were based were detected using CUSUM in order to reduce the search space and convert continuous data stream into discrete data stream. In the third step, the correlation rules were generated. The proposed method used sampling techniques to determine a few candidates to work with in order to speed up the search.

Wang et al. [99] proposed a technique for similarity search and discovery of patterns in time series. Using a vector quantization technique for dimension reduction and a symbolic representation of time series, they apply a string matching technique, LCSS (longest Common subsequence), to compare time series of different lengths. The authors address the case of a nearest neighbour of the input query with a threshold given for the distance measure. The proposed solution partitions each sequence into equivalent length segments and uses vector quantization to represent each segment into a codebook. The similarity measure is then carried out using an extended version of the LCSS that will allow to compare sequences of different lengths.

Zhang et al. [111] extend the clipping technique [81] to propose a technique for

correlation analysis in stream time series. Using binary sequences to dynamically represent streaming time series, a similarity search and clustering operation is ran on incoming streams. The article focused on fast correlation analysis in a large number of stream time series and proposed a technique called HBR (Hierarchical Boolean Representation). The technique processes a large number of incoming stream series, transform them into boolean series, before to check their pairwise correlation values against given correlation thresholds. A candidate set is then identified as containing all series for which their boolean versions are correlated above the given thresholds. An extension of the Pearson similarity measure is subsequently used to compute the correlation value between original time series selected as part of the candidate set.

The problem of exploring and identifying correlations among multidimensional arrays of data stream (particularly between two such arrays in environments with limited resources) was explored by Wang et al. [100]. For this, they propose to use an improved version of the standard canonical correlation analysis technique (CCA). When used in its standard form CCA (or NaiveCCA) does not perform well for multidimensional time series. This technique , termed ApproxCCA [100] uses an incremental computation paradigm and low rank approximation technique based on unequal probability sampling to reduce the dimensionality of the product matrix (composed by the sample covariance matrix and sample variance matrix). The proposed changes were mainly carried out in two steps: an incremental computation performed on the variance and covariance matrices, and a low-rank approximation conducted on the product matrix to obtain the canonical eigenvalue and the canonical correlation eigenvector.

In recent years we witness more research effort looking to apply real world situations [36, 80] where data can be efficiently processed with its imperfection (e.g. uncleansed). An example of such trends is the work in [36] by Hu et al. which is intended to be robust for environments with irrelevant and missing data scenarios. The technique is based on the general techniques of weighted voting and Bayesian classification and, extends the techniques of dictionary-based classification. More precisely, a framework to classify multi-dimensional time series by weighting each classifier's track record is proposed. The classifier's track record weights are themselves based

on each stream's previous track record on the class it is predicting at the moment, but also on the distance from the unlabeled object.

Raptis et al. [80] presented a real-time wireframe for skeletal motion classification. The key components of this technique include three aspects: an angular representation of the skeleton, to render recognition more robustness for noisy input, a cascaded correlation-based classifier for MTS data with a distance metric based on dynamic time warping to evaluate the difference in motion between an acquired gesture and, an oracle for the matching gesture. While the first two key components can be re-used for other skeletal motion classification scenarios, the oracle for the matching gesture is specifically tailored for a known canonical time based musical beat. The technique provides a framework that serves to guide individuals during exercise sessions (in this case dance instead of rehabilitation).

While some progress has been noted in the area of streaming MTS analysis and search, it is still in its infancy and more advanced techniques are needed to seamlessly and adaptively cater to streaming data in large data frameworks, regardless of its arrival rate. In this thesis work, we developed efficient and scalable techniques for MTS data at rest that we plan to extend to streaming data frameworks in our future work.

2.2.2 Multivariate Time Series Similarity Search Techniques for Data at Rest

The second group of techniques concerns MTS similarity search for data at rest. While techniques in this group are today increasingly investigating big data challenges pertaining to MTS with large number of variables and instances, early techniques, investigated multivariate time series with fairly small number of variables. They often relied on indexing techniques. Those early techniques included the work of Vlachos et al. [98], investigating similarity search in the trajectories of moving objects in two to three dimensional spaces with different orientations. The technique used an indexing scheme that leverages the hierarchical tree of a clustering algorithm for nearest neighbor queries. The technique afforded among other things, a time

series normalization strategy, a mapping of trajectories to space that would be robust against rotations and translation scaling, an elimination of the noise that results from vertical shifting by mapping the spacial coordinates of the trajectories to sequences of arcs and angles pairs prior to normalization. It also accommodated assessing the similarity between variable lengths time series data.

In a similar framework, Chen et al. [14] also studied MTS similarity in relation with moving object trajectories. Data from such frameworks is known to be carrying much noise, hence this technique like many others investigated ways to devise a technique that would be robust against noisy data. An extension to the Edit or Levenshtein distance was introduced as Edit Distance on Real sequence (EDR). It allowed to assess the dissimilarity between two objects by counting the minimum number of edit operations required to transform one object into the other, while handling local time shifts and assigning penalties to the unmatched parts. The distance measure was combined with different implementation methods of three pruning techniques, known as mean Q-gram techniques to improve the retrieval process.

The other strategy within this group of techniques is attempting to reduce dimensionality and transform the MTS prior to measuring the similarity for relatively larger number of variables. This latter strategy seems more suited for multivariate time series with large number of variables due to the fact that, when careful selected, the dimension reduction techniques often allow for more efficient data processing. In this framework, Yang et al. [103], proposed a similarity measure for MTS, based on PCA, called Eros (Extended Forbenius Norm). The authors propose to represent two MTS as two matrices (e.g. A and B), whose PCA are computed to uncover their respective eigenvalues and eigenvectors. The eigenvalues and eigenvectors are then subsequently used to uncover the dataset weight vector w and measure the similarity between the two matrices. Provided with the two original matrices and the dataset weight vector w, the technique assesses the similarity between the two datasets by computing Eros(A,B,w)=$\sum_{i=1}^{n} w_i| < a_i, b_i > |$, where $< a_i, b_i >$ is the inner product of a_i and b_i. This proposed similarity measure does not satisfy the triangle inequality, but can be provided with an upper and lower bounding by using the weighted Euclidean distance. In combination with Eros, a search based K nearest neighbors (KNN) for

MTS is proposed. Experimentally, the proposed solution outperforms the Euclidean distance(ED), Weighted Sum SVD (WSSVD), Dynamic Time Warping (DTW), and the PCA similarity factor (SPCA) in terms of precision and recall. It also presents a better time complexity than the Euclidian distance and DTW.

Yang et al. further investigated the idea in [104] by exploring how the stationarity of the time series might impact the proposed technique. The time series were first rendered stationary before processing. A time series is considered stationary if its correlation is stable, i.e. if its statistical properties such as the mean and the correlation coefficients, do not change over time. Applying Eros on brain stimili time series rendered stationary showed a performance improvement in precision and recall of over 24%.

Karamitopoulos et al. [51] proposed a MTS similarity search technique based on reducing the MTS data set size and dimensionality using a principal component analysis (PCA) signature prior to measuring similarity. The idea explored in the paper is the recognition that if two MTS are similar, their PCA representations will be similar in some way as well. An interesting aspect of this similarity search technique is that it does not directly rely on applying each time a intensive computation of the PCA to query frequently arriving object in order to match them to the most similar objects in a database or classify them into predetermined classes. Rather, the resource intensive computations of the PCA are conducted only once to build up a database of PCA signatures, which will be subsequently used to allow the quicker identification of a query objects most similar correspondent in the database.

Tanaka et al. [94] devised a technique for finding motifs in MTS. This technique, like many in this group, also relies on the recognition that if two multivariate time series are similar, their PCA representations will be similar in some way [51, 4]. The technique first relies on dimensionality reduction and transforms the MTS to a UTS by applying a standard PCA. It retains the first principal component to be used in the subsequent steps. The dimensionality reduction is followed by a transformation of the data to a sequence of symbols using SAX(Symbolic Aggregate Approximation) [67] an extension of the PAA (Piecewise aggregate approximation) [107, 57] technique. The technique subsequently looks to find motifs using the MDL (Minimum Description

Length) principle on all dimensions of the MTS. While, this technique provides good results for relatively low number of variables, and cases where the first principal component carries a large amount of the explained variance, computing the standard PCA algorithm on a large matrix normally requires O(min (n^2m, m^2n)), which can be prohibitively expensive in large dataset environments. Techniques transforming the multivariate time series to the univariate domain often yields better results as they tend to retain more characteristics from the original data [86, 82].

Banko et al. [4] proposed to use the correlation based dynamic time warping (CB-DTW) technique to find correlations between MTS for applications in MTS recognition. This technique goes through three steps to assess the similarity between MTS. In the first and second steps, the time series of the given database and the query time series are segmented respectively; in the third step the distance between the two is assessed. The segmentation strategy is the same for both the query and the time series contained in the database. A key aspect in this segmentation process is the need for homogeneous segments, hence correlation was used for that purpose. The segments are obtained by bottom-up segmentation, where every element of the whole UTS within a given MTS is handled as a segment and the costs of the adjacent segments are calculated (using special, PCA related cost). Subsequently two segments with the minimum cost are to be merged. This being a top down strategy, The merging and cost calculation of adjacent segments continues until some set goal is reached. After the MTS are segmented, the third step consists of assessing the DTW dissimilarity between the query and the time series from the database. For that purpose the PCA similarity factor (SPCA) is uncovered on each segment to feed the dynamic time warping (DTW) local dissimilarity function. In combing DTW and PCA based similarity measures, the technique preserves the internal correlation that exist between variables, and allows for advantages such as a robustness against phase shifts of the time axis, or accommodating differences in sampling rates. The technique outperformed peer techniques on the 2004 competition database of signature verification and the AUSLAN datasets.

In applications to classifications, Esmael et al. [22] proposes a technique for MTS classification. The technique extends the Symbolic Aggregate Approximation (SAX)

technique [67] using three new string symbols (U, D and S) to represent the trend of the time series (UP, Down, Straight). First each variable within the multivariate time series is divided into a sequence of smaller segments by sliding a window incrementally across the time series. Then the algorithm transforms the numerical values of each variable in the given time series into a sequence of ⟨value, trend ⟩ pairs. While a good level of precision can be expected from this technique because the quantization happens directly on the data, it will not easily scale to large multivariate dataset environments, where the number of variables are expected to be sizably larger.

Also in a clustering/classification frameworks, Spiegel et al. [92] proposed an approach that can be generalized for contextual pattern detection in multivariate sensor data. The technique consisting of three steps: feature extraction based on important time series characteristics to devise learning models, uncovering internally homogeneous time intervals and change points through segmentation, and finally clustering and/or classifying time series segments into the sub-population to which they belong to, based on contextual similarity. The segmentation procedure relies on a PCA based segmentation technique derived from a time-varying multivariate data segmentation method. The latter is known to be robust for signal processing and complex drive maneuvers that consist of multiple consecutive time series segments. Examples of such maneuvers include for instance "stop and go at traffic lights or overland drives" [92].

In this trend of research as well, much progress has occurred in recent years. However there is still a need to improve on many aspects of the current available techniques. As we will see, in Chapters 3, 4, 5, and 6, our introduced techniques provide improvements on precisions and recall, memory space usage and processing time when compared to peer techniques.

2.3 Summary

Multivariate Time series analysis has grown in importance, relevance and popularity in recent years. Traditional methods for uncovering meaningful patterns are no longer suitable in today's high dimensional data processing frameworks. In this Chapter, we

introduced the background and literature pertaining to data reduction and similarity search in MTS. Although good progress has been achieved, more work is required to improve on many aspects of the current available techniques. For instance, much of the current literature for similarity search is still focused on fairly low size multivariate problems with less than 150 variables. In addition, even finding suitable ways to reduce and represent time series in the preprocessing stage is still an open problem.

We however believe that better results for MTS analysis and search in large data frameworks can be achieved by combining some important aspects among which the following three:

- understanding that high performance data driven discovery or unsupervised learning in MTS is becoming a necessity in large data frameworks.
- leveraging powerful tools and techniques such as the Principal Component Analysis(PCA), Randomized Matrix Theory aided by a heightened computational power,
- adopting a multidisciplinary approaches to allow for a richer and broader research frameworks.

Our contributions in Chapters 3, 4, 5 and 6 follow those lines of thoughts. In the next Chapter, we introduce an efficient dimensionality reduction, and more precisely feature selection technique using PCA and based on unsupervised learning.

Chapter 3: Feature Selection

In this chapter, we introduce the Weighted Scores(WS) technique, an unsupervised learning technique to identify the top-k discriminative features for multivariate time series(MTS).

The Weighted Scores technique uses statistics drawn from the Principal Component Analysis (PCA) of the input data to leverage the relative importance of the principal components along with the coefficients within the principal directions of the data to uncover the ranking of the features. In what follows, we review the background and preliminaries in Section 3.1, discuss the related work in Section 3.2, and introduce the technique in Section 3.3. Section 3.4 presents the performance evaluation. A summary, our concluding remarks and future directions are presented in Section 3.5.

3.1 Background and Preliminaries

Innovation and advances in technology have led to the growth of data at a phenomenal rate. Paradoxically, the existing MTS data reduction, analysis and mining techniques do not scale well to its current challenges. Among those challenges, the presence of noise and redundancies in high dimensional data for many practical applications makes it difficult to uncover important patterns. Hence, most pattern recognition tasks rely on dimensionality reduction through feature extraction or feature selection as crucial preprocessing steps, for reasons of efficiency and interpretability, for a better understanding of the underlying processes that generated the data, but also to build

a framework that allows downstream pattern recognition tasks such as classification, clustering or similarity search to perform more efficiently.

While feature extraction and selection both ultimately allow to use a reduced set of features to achieve higher or similar accuracy results as using all features, they differ in terms of strategies. On one hand, feature extraction relies on transforming existing features/variables into a lower dimensional space, and creating a new and reduced number of features from original features. The new features are conceived so to have the largest possible variance, since the percentage of explained variance retained for the new variables indicates the amount of information retained within the reduced data. The larger the variance retained, the lower will be distortion at reconstruction. Feature extraction techniques often can uncover the new embedding space in linear time, making them preferred in terms of computational complexity. Unfortunately, the transformation process in feature extraction methodologies can render the newly extracted features difficult to interpret.

On the other hand feature selection (also known as variable or attribute subset selection), identifies a subset of most relevant features from the original set to be used for subsequent processes. The difference in reduction strategies makes feature selection preferred as a dimensionality reduction method for practitioners in cases where the interpretability of the reduced subset needs to be maintained with respect to the originally acquired features, or when there are far more features than samples [27]. The feature selection process is however a combinatorial optimization problem, which is NP Hard [72, 15]. To uncover the needed subset in reasonable time, most techniques in this area rely on heuristics. In our case, we rather rely on matrix decomposition. We present a simple yet effective and scalable unsupervised feature selection technique based on statistics drawn from the Principal Component Analysis (PCA) [41] of the input data. While this technique leverages the desired properties of the PCA, it retains the results interpretability of the results by combining the advantages of feature extraction and selection techniques.

An important aspect of this technique is the recognition that not only the coefficient within the principal directions are crucial in identifying the importance of variables, as seen in major related work [41, 109, 18, 90] but also more importantly

each principal component should be weighted with its relevance during assessment of the variable's weight.

Our empirical evaluation in a large number of application domains using numerous benchmark datasets indicate our proposed technique improved performance in terms of accuracy, speed, and scalability.

3.1.1 Principal Component Analysis and Singular Value Decomposition

In this section we review the PCA and SVD techniques which form the basis for our proposed techniques in Chapters 3, 4 and 5.

The PCA and SVD are orthogonal linear transformations and matrix decomposition techniques that project the original dataset in a new coordinate system where the directions are pairwise orthonormal. A main advantage of these techniques in our work is that they guaranty the uncovering of an optimal reduced embedding with minimal approximation error, and hence retains the crucial underlying structure of the original data. The PCA and SVD are amongst the most efficiently computable techniques and powerful tool of choice for reduction of high dimensional data. Besides, these two techniques decrease redundancy and noise, highlight relationships between the variables, and reveal patterns by compressing the data. They help identify similarities and dissimilarities as well. Using these two techniques on raw MTS data often requires some preprocessing such as mean-centering and scaling to adjust values measured on different scales to a relatively common scale, since PCA is a variance maximizing technique. In addition, some conditions [93] on the raw input data have to be met for the results from the PCA/SVD to be numerically stable and valid. These conditions are primarily as follows:

- The relationships between variables must be linear.
- There should be "adequate" redundancies amongst the variables through correlation to allow summarization of the variables into a rich, reduced set of components.

- The sample data size must be adequate compared to the number of variable.
- No significant outlier must be present in the data since otherwise bias may be introduced in the computation of the PCA for such data to have more weight than the remaining data.

Many large datasets occurring in practice meet these conditions and exhibit suitable characteristics. Due to their sizes, such frameworks often present relationships between its variables, and are also much prone to redundancies and noise. In cases where the data fails to meet these conditions, additional preprocessing may be required to help remediate the issues, explained as follows.

For the first condition, the relationships between variables may be visualized on a matrix scatterplot, or by randomly sampling variables in large scale settings. When the relationships are non-linear, transformations can be carried out using non-linear methods such as linear regression, exponential models among others to achieve linearity.

For the second condition, the adequacy of the redundancy in the dataset can be uncovered by using the Bartlett's test of sphericity. While one principal component will suffice in the cases where the variables are perfectly correlated, in the case where they are orthogonal (uncorrelated), having to retain a number of component equal to the number of variable is likely to be expected.

For the third condition, the suitability of the sample size can be tested using the Kaiser-Meyer-Olkin (KMO) technique. This condition is the most difficult to fulfill in large scale environments because many datasets, particularly in health sciences are "sample starved". In cases where there are far more variables than samples, a number of methods have been proposed to test for sampling adequacy. The technique proposed in [43] to assess the stability of the principal components is particularly suitable for cases where we have a small sample size compared to a large number of variables.

While the PCA is often explained through an eigen-decomposition of the covariance matrix (A^TA) for an original data matrix $A_{n,m}$, it can be performed through the SVD of the data matrix. In our work, we consider the latter.

The relationship between the SVD and the PCA can be uncovered as follows: Let A be an $n \times m$ data matrix and $C = A^T A/(n-1)$ be its $m \times m$ covariance matrix. C is a symmetric matrix that can be diagonalized as $C = VDV^T$ where V and D are respectively the matrices of eigenvectors and eigenvalues λ_i(on a diagonal matrix in decreasing order of importance, where i = 1, 2, ..., m). The principal components, also labeled as PC scores are obtained by projecting the original data on the eigenvectors, also called principal directions or principal axes. By doing so, we obtain the transformed variables in AV.

The singular value decomposition of A can also be expressed as $A = USV^T$ where S is the diagonal matrix of singular values s_i, where i = 1, 2, ..., m. We can then see that, with $C = A^T A/(n-1)$ and $A = USV^T$, C can be rewritten as $C = VSU^T USV^T/(n-1)$, hence $C = V\frac{S^2}{(n-1)}V^T$. This shows that the right singular vectors V are principal directions and eigenvectors of the covariance matrix can be uncovered from the singular values by using the relationship $\lambda_i = s_i^2/(n-1)$. The principal components can also be uncovered from both AV and US since $AV = USV^T V$, hence as $AV = US$. Each principal component points in the direction of maximal variance and excludes the variance already accounted for by the previous principal components.

The Principal Component Analysis based on the Singular Value Decomposition of a data matrix can be described as follows:

Let $A_{n,m}$ be a matrix with n dimensions and m variables, and k be the dimension of the space in which we wish to embed the data. Using a Singular Value Decomposition of the matrix, PCA returns the top k left and top k right singular vectors of A. It subsequently projects the original data on the k-dimensional subspace spanned by the chosen column singular vectors.

Definition 3.1. *(Singular Value Decomposition) Let A be an $n \times m$ data matrix with r as its rank. The singular value decomposition (SVD) of A is the factorization $A = USV^T$, where:*

- *U is a column-orthonormal $n \times r$ matrix whose columns are the eigenvectors of AA^T,*
- *S is a diagonal $r \times r$ matrix of the singular values s_i for A, otherwise related*

to the eigenvalues λ_i *of the covariance matrix* A^TA *by* $\lambda_i = s_i^2/(n-1)$*, where* $\lambda_1 \geq \ldots \geq \lambda_r \geq 0$*, and,*

- *V is a column-orthonormal r×m matrix whose columns are the eigenvectors of* A^TA*.*

For large scale settings, the randomized PCA can similarity be computed using a randomized SVD. While both the standard and randomized SVD techniques allow to uncover the best rank-k approximation of the original matrix, the standard approaches are computationally expensive, often with O(min(n^2m, m^2n)) time complexity. Alternatively, the randomized techniques have proven to maintain high accuracy relative to the standard techniques while providing far better computational cost in high dimensional frameworks [30, 29, 17]. These randomized techniques present different resolution strategies but all often rely on the key paradigm that the original matrix can be intuitively and carefully approximated, hence, the few needed principal components uncovered without having to thoroughly compute the full SVD.

From this Singular Value Decomposition, the new matrices that are of interest for us are matrix S, which contains the singular values, and matrix V^T, whose rows represent the right eigenvectors. Often, a rank-k approximation of the dataset ($A_k = U_kS_kV_k^T$) works well because many datasets in practical applications have structures such that only the first few principal components are non-negligible.

To identify the number k of principal components to retain we use the relative percentage variance criterion to translate the amount of variance we wish to retain in the data to the number of principal components.

Algorithm 3.1 summarizes the steps in uncovering the number of principal components to retain. Algorithm 3.2 provides the steps within our proposed technique.

3.1.2 Problem Formulation

A MTS $A_{n,m}$ of n instances for m variables/features can be represented as an $n \times m$ matrix A (shown below) in which $a_{i,j}$ is the value of variable v_j measured at timestamp i, for $1 \leq i \leq n$, $1 \leq j \leq m$.

$$A_{n,m} = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,m} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,m} \end{bmatrix}$$

We are interested in the problem of unsupervised feature selection that can be framed from a general perspective as:
Let A be a data matrix of n instances and m features. The goal is to find the most informative features within the data matrix, without relying on class labels to guide the search. The expectation is to preserve the intrinsic structure of the original data as represented by all features when the data is represented using the selected subset of features.

The problem of unsupervised feature subset selection is also often considered as an instance of the column subset selection (CSS)problem [13, 10] defined more formally as follows:

Definition 3.2. *(Column Subset Selection) Let A be an $n \times m$ data matrix of real numbers and $k < m$ be the number of columns we look to retain to form the matrix C of dimensions $n \times k$. A column subset selection operation uncovers the k best columns of A such that the residual $\|A - CC^{\dagger}A\|_{\xi}$ is minimized over all possibles combinations of k columns out of m choices for matrix C. Here, $CC^{\dagger}A$ denotes the projection onto the k-dimensional space covered by the columns of C, $\xi = \{2, F\}$ denotes the spectral norm and Frobenius norm respectively, and $C^{\dagger}$ stands for the Moore-Penrose pseudo-inverse of C.*

Essentially in the CSS problem, we seek to retain C columns out of A that best capture as much as possible of A with respect to the spectral or Forbenius norm.

3.2 Related Work

Feature selection techniques can be classified under two main categories: supervised and unsupervised. Supervised feature selection methods such as Fisher Score [20] or

ReliefF [61] usually evaluate the feature importance by uncovering the dependence between feature and class labeled. These techniques assume that the data is labeled, while this can't be assumed for high dimensional frameworks where obtaining labels can be very expensive or impossible. Unsupervised techniques such as the Leverage Score sampling [42] use all data points to uncover the hidden structure from unlabeled data and infer a relation that allows to best categorize the data. Selecting features in such scenarios is a much more difficult problem and can become intractable (NP-Hard) very fast since we are dealing with a combinatorial optimization problem [72, 15].

Another categorization of feature selection techniques is based on whether it is a Filter or Wrapper method. Wrapper methods search for an optimal feature subset adapted to the specific mining algorithm, which in turn enhances the performance of the specific learning task. Filter methods on the other hand analyze the intrinsic properties of the original data, and select highly-ranked features according to some defined criterion such as variance, entropy, smoothness, density or reliability [27]. Filter methods are known to be more scalable, effective in computation time, and robust to overfitting.

The literature pertaining to feature selection in general is vast and can be placed back to about six decades ago [6, 8]. Early techniques where often based on probabilistic strategies [37]. Recent literature is increasingly directed towards strategies looking to address the challenges that result from high dimensional frameworks [6]. More specifically, and although scarce, some effort has been made to investigate feature selection in MTS [108, 32, 16, 106]. Those techniques are unfortunately often supervised relying on wrapper methods, which can often be expensive in large dataset frameworks. There is a need to develop unsupervised techniques better suited for time series data in such frameworks.

Recent directions in unsupervised feature selection have included techniques that looked to identify features that are best at preserving some implied cluster structure within the dataset. In such cases, the strategy is often to maximize some clustering performance, while maintain the cluster coherence after the data is represented using only the selected variables [101].

Other techniques looked into features that favored locality preserving features.

Such techniques often look for criteria such as the similarity between data instances and construct a k nearest neighbor graph, in order to retain the features that best preserve the graph structure [73].

Another set of techniques looked to uncover the features that would be most representative of the data. Among those, the variance maximizing methods are known to be among the preferred ones due to their simplicity and efficiency in conserving the geometrical structure of the data space. In that particular direction, many techniques rely on the PCA for efficient unsupervised feature selection. This is primarily due to the fact that the PCA is known to be one of the most efficiently computable techniques for linear dimensionality reduction. Jolliffe [41] studied different algorithms using PCA for unsupervised feature selection, where the features are associated with principal components based on the absolute value of their coefficients and selected or discarded depending on weather they belong to the first few or last few principal components. Particularly Jolliffe [42] proposes a method that consists of sampling columns from the original matrix A that correspond to the largest leverage scores $\ell_i^{(k)}$, for some rank $k < rank(A)$. The leverage score of the i^{th} column of A $\ell_i^{(k)} = \| [V_k]_{i,*} \|_2^2$ for $1 \leq i \leq n$ is computed as the squared Euclidean norm of the i^th row of the matrix V_k containing the top right singular vectors (otherwise known as principal directions).

A randomized version of the technique was proposed in Drineas et al. [18], where for some rank $k < rank(A)$, a probability distribution over the i^{th} columns of A was defined as $p_i = \ell_i^{(k)}/k$, where $\Sigma_i \ell_i^{(k)} = \|[V_k]\|_F^2 =$ k and $\Sigma_i p_i = 1$. It retains C number of columns sampled from A with probabilities proportional to their leverage scores.

In [109], Yoon et al. determines the contribution of each feature (or band) to the new features (principal component score image bands in this case of a hyper-spectral imaging application). The squares of the principal direction coefficients at each band are computed and normalized to the sum of 1 according to the equation: $W_k(i) = P_k(i)^2 / \Sigma_{i=1}^n P_k(i)^2$, i $\in \{1, ..., n\}$, where $W_k(i)$ is a weighting factor of the i-th feature of the k-th principal component, and $P_k(i)$ is the principal direction(eigenvector).

All these technique have been quite successful in many application domains and fundamentally rely on the basic idea of using PCA for feature selection by choosing

columns according to the magnitude of the coefficients on the principal directions. Those coefficients are understood as the weights (or the amount of contribution) of each input variable to the corresponding principal component.

However, an important aspect they overlook is the recognition that by additionally factoring in the relative importance of each principal component prior to computing the feature weight, the accuracy in identifying the most relevant features is highly improved.

3.3 Weighted Scores

In this section, we formally define our proposed technique, and describe its underlying intuition.

Definition 3.3. *(Weighted Scores) - Let V_k contain the top k right singular vectors of matrix $A \in R^{n \times m}$ with rank $r = rank(A)$ s.t. $r \leqslant min\{n, m\}$ and $k \leqslant r$. Then, the (rank-k) weighted score of the i-th column of A is defined as*
$wS_i^{(k)} = |\Sigma_{j=1}^{k} w_j t_{i,j}|$, for $1 \leqslant i \leqslant m$, where:

- *$w_j = \lambda_j / \Sigma_{z=1}^{r} \lambda_z$ is the fraction of variance carried by the j-th column in $[V_k]$, for $1 \leqslant j \leqslant k$*
  $\lambda_j = \sigma_j^2/(n-1)$ is the variance corresponding to the j^{th} singular value, consequently to the j^{th} column of $[V_k]$, and $\lambda_1 \geq \ldots \geq \lambda_r \geq 0$, and
- *$w_j t_{i,j}$ is the j-th column entry of the i-th row $[wV_k]_{i,*}$ of the weighted matrix $[wV_k]$.*

The goal is to identify the best subset of k features from the original data matrix, which would retain important characteristics while relying on the PCA. Our proposed technique approaches the problem from a numerical analysis perspective and utilizes the information contained in matrices S and V resulting from the Singular Value Decomposition of A expressed as USV^T. Here, S carries the singular values in decreasing order, and the columns of V are the right singular vectors also arranged in decreasing order of importance. Those right singular vectors in V are known as the

eigenvectors of the covariance matrix $A^T A$. Hence, the principal components carry in decreasing order an explained amount of variance from the data.

Intuitively, if we look to recombine the principal components into a new framework, it will be crucial to factor in their importance/weight in relation to the provided MTS dataset. The importance/weight is reflected through the proportion of explained variance retained by the specific j-th principal component labeled $w_j = \lambda_j / \Sigma_{z=1}^{r} \lambda_z$ in our technique. To obtain the weighted-matrix wV_k, each component within the retained matrix of eigenvectors V_k is multiplied by its corresponding weight w_j, where $j = 1, 2, \ldots, k$.

In addition, the entries in each eigenvector provide the regression coefficient of its corresponding principal component, which in turn is expressed as a linear combination of all the variables from the original matrix. More precisely, the coefficient of the i^{th} feature from the original matrix uncovered through PCA is expected to be the i^{th} entry of the eigenvector.
The first k principal components can be expressed as follows, where $X_1, ..., X_m$ are the original variables within the data matrix A.

$$a_{1,1}X_1 + a_{1,2}X_2 + a_{1,m}X_m = PC_1$$
$$a_{2,1}X_1 + a_{2,2}X_2 + a_{2,m}X_m = PC_2$$
$$\ldots$$
$$a_{k,1}X_1 + a_{k,2}X_2 + a_{k,m}X_m = PC_k$$

The largest (in absolute value) coefficients of the linear combination identify the most relevant variables for the given principal component and reveal otherwise hidden, implied relationships. We must note that we take into account negative coefficients, since the intent is to uncover the aggregated effect of the variable within the given dataset.

The (rank-k) weighted score of the i-th column of A is then computed as $wS_i^{(k)} = |\Sigma_{j=1}^{k} w_j t_{i,j}|$.

The greater variance along with higher eigenvector's coefficients (positive or negative) are considered as the most important factors when selecting a variable in the

Algorithm 3.1 - Uncover the number k of PCs to retain

Input: A $\in R^{nxm}$, θ (cumulative variance explained)
Output: k, the number of principal components to retain.
begin

1: Uncover fraction of total explained variance
2: f(k) $\leftarrow \Sigma_{z=1}^{k}\lambda_z/\Sigma_{z=1}^{r}\lambda_z$ for all z = {1, . . . , r}
3: Choose the smallest k so that f(k) $\geqslant \theta$ and retain that number of k eigenvectors to keep explained variance θ in the new embedding.
4: return k

end

ranking process. As illustrated in line 7 of Algorithm 3.2, the features are ranked in decreasing order of importance to allow for an easier selection process.

The technique leverages advantages from both feature extraction and selection techniques. It relies on SVD and PCA to find an optimal embedding in the lower subspace and subsequently identifies the contribution of each original variable.

The *Weighted Scores* (WS for short) algorithm3.2 summarizes the steps in our proposed technique.

3.4 Performance Evaluation

To evaluate the effectiveness of the Weighted Scores technique, we implemented the code in Matlab and conducted numerous experiments on benchmark datasets. We used a PC configured with Intel Quad core i7 2.00GHz CPU, 8GB RAM, and running Windows 7.

3.4.1 Benchmark Datasets

The experiments were ran on over fifty synthetic and real time series benchmark datasets, from many domain applications, including UCR Time Series Classification Archive [55], UCI repository [66], FMRI datasets from [44, 63, 19], and financial market datasets [78].

Algorithm 3.2 - Weighted Scores (WS)

Input: Matrix $A \in R^{n \times m}$ of rank-r, θ (cumulative variance explained)
Output: $S_r \in R^{n \times c}$ which has the top c most representative ranked features of A.
begin

1: Compute the Singular Value Decomposition
$[U, S, V^T] \leftarrow SVD(A)$
2: Compute the proportion of variance carried by each component
For j $\leftarrow$ 1 to r
$\lambda_j \leftarrow (s_j^2/(n-1))$, where $s_j \in S$
end for
3: Identify the number k of principal components to retain
k $\leftarrow$ Algorithm 3.1(A, θ)
4: $M \leftarrow V_k$
5: Build the weighted matrix $[wV_k]$
For j $\leftarrow$ 1 to k
$w_j \leftarrow \lambda_j / \Sigma_{j=1}^{r} \lambda_j$
$[wV_k]_{*,j} \leftarrow w_j * [M]_{*,j}$
end for
6: Compute the weighted score for each variable
$wS_i^{(k)} = |\Sigma_{j=1}^{k} w_j t_{i,j}|$, for all i = {1, 2, ..., m}.
7: Sort the variables according to their weights:
$C^{n \times m} \leftarrow \mathrm{w}S_1^{(k)} \geq ... \ \mathrm{w}S_i^{(k)} \geq ... \geq \mathrm{w}S_m^{(k)}$
8: Select the top-c features
$S_r \leftarrow C^{n \times c}$

end

Table 3.1: Benchmark Datasets

Datasets	Features	Instances	Classes
Arrhythmia	279	452	16
CorAl	6	32	2
Madelon	500	1800	2
Mallat	1024	2345	8
Gisette	5000	6500	2
Ionesphere	34	351	2
Iris	4	150	3
Reuters	5080	1806	2
Soybean	35	47	4
S&P	500	1153	4
Synthetic Control Chart	60	600	6

In an attempt to uncover how our proposed technique would fair against other techniques on classical, well-known datasets specifically designed for feature selection problems, we conducted experiments on datasets other than time series. For that, we used 5 high dimensional data sets from [27], 25 small benchmark datasets from [26], and 12 small datasets used in [11].

Our aim has been to cover a wide variety of datasets with different and challenging characteristics in feature selection. Table 3.1 and what follows provide more details about the datasets discussed in this section.

The Arrhythmia dataset [66] was generated to distinguish presence and absence of cardiac arrhythmia.

The CorAl dataset [66, 60] allows to test feature selection in the presence of highly correlated features, a nonlinear concept and irrelevant features. Its first four features completely determine the target concept: $(A \wedge B) \vee (C \wedge D)$. The fifth feature is irrelevant, while the sixth is highly correlated with a 25% error rate (matches the class label 75% of the time).

The Fisher Iris dataset [66] is based on Fisher's linear discriminant model, distinguishing iris species from each other. While one of the classes is linearly separable from the other two, the other two are not linearly separable from each other.

The Madelon dataset [27, 66] presents Gaussian clusters positioned on the vertices of a hypercube and labeled randomly. It is generated with 20 relevant and 480 noise features. The test set was artificially constructed with 500 variables, 2 classes, and 1800 samples to illustrate the difficulty of selecting a feature set when no feature is informative by itself, and all the features are correlated with each other [27].

The Mallat dataset [55] is a synthetic dataset known to have difficult signal patterns. It was used to evaluate the error rate in applying Classification Regression Tree (CART) to the reduced size data for classifying process fault types.

The MLSP 2014 Schizophrenia dataset [63] was made available as part of an official competition of the IEEE International Workshop on Machine Learning for Signal Processing (MLSP 2014). The information was used to select features that enhance diagnosis on schizophrenic individuals.

The Gisette dataset [27, 66] is high dimensional and generated for a handwritten digit recognition application. It was used to assess how well different techniques can separate the two confusable classes: four and nine.

The Ionosphere dataset [66] consists of radar data collected by a system in Goose Bay, Labrador. Radar signals targeted free electrons in the ionosphere, while labeling radars showing evidence of some type of structure in the ionosphere as "Good", and those that do not as "Bad".

The Reuters dataset [66] is frequently used in text categorization applications. It consists of documents that appeared in Reuters newswire in 1987. Each document was then manually categorized into a topic among over 100 topics.

The Soybean Small dataset [66] is often used in multi-class classification problems to predict problems with soybean crops from crop data.

The S&P dataset [102] consists of adjusted daily closing stock prices gathered from Yahoo for 500 stocks that make up the S&P 500 index during 1153 days (from Jan 2, 2003 to Aug 1, 2007). An 1153×427 matrix was generated and mean-centered for processing; 73 columns with missing values were discarded. Labels were generated to also evaluate the performance of unsupervised techniques as part of the study. We generate 4 classes, one for each time period of the year where data instances occur (respectively 1^{st} quarter, 2^{nd} quarter, 3^{rd} quarter, 4^{th} quarter).

Synthetic Control Chart Time Series dataset [66] consists of synthetically generated control charts data to help define the notion of similarity between time series. The control chart patterns are time series that show the level of a machine parameter plotted against time and generated according to the process described in Alcock and Manolopoulos [1].

3.4.2 Peer Techniques

We compared our technique against the following seven supervised and unsupervised techniques:

- **Clever** [105] is an unsupervised technique which uses PCA and leveraging a Common Principal Components (CPCA) between MTS.
- **FSPCA** [90], feature selection using principal component analysis (FSPCA) is unsupervised. It exploits results from a PCA of the covariance matrix to evaluate the significance of each feature component.
- **Fisher Score(FS)** [20] is a supervised technique that seeks a subset of features, such that the distances between data points reflect memberships to classes.
- **Leverage Score(LS)** [42] is unsupervised and relies on PCA. It samples features from the original matrix that correspond to the largest leverage scores.
- **Leverage Score Sampling(LSS)** [18] is an unsupervised technique, considered as a randomized version of the technique proposed in [42]. It samples features corresponding to probabilities proportional to its largest leverage scores.
- **Max-Relevance Min-Redundancy(mrmr)** [77] is a supervised technique which selects good features according to the maximal dependency criterion based on mutual information.
- **ReliefF(RfF)** [61] is a supervised technique which estimates the quality of attributes according to how well their values distinguish between instances that are near each other.

Our proposed technique leverages variance as a criterion to identify and rank relevant features. Other techniques based on variance and here compared to our technique include Fisher Score [20], Leverage Score [42, 18], Clever [105], FSPCA [90]. In these techniques either the highest ranked features are chosen [42, 105, 20, 90] or features are randomly sampled, with a probability proportional to their importance [18]. We also compared our results in this section with techniques based on other criterions such as dependence (e.g. Max-Relevance Min-Redundancy(mrmr) [77]). The proposed technique outperforms the other techniques in terms of speed and accuracy on a large number of datasets as shown in the empirical results; details of which follow.

3.4.3 Evaluation and Results

We designed sets of experiments, among which three discussed in this section.

3.4.3.1 Ranking Features and Minimizing the Residual

For this set of experiments, we evaluate the performance of our Weighted Score (WS) technique against other techniques on a large number of datasets. For each dataset, we retain the set of k most relevant features and subsequently evaluate the different methods according to their ability to minimize the residual $\|A - CC^{\dagger}A\|_{\xi}$. Here, as seen in Definition 3.2, A is the original data matrix with all features, C is the matrix build out of the k identified features and, $\xi = \{2, \mathrm{F}\}$, where the value 2 denotes the Euclidean norm and F denotes the Forbenium norm. The goal is to identify which technique provides the best set of features.

In minimizing the residual $\|A - CC^{\dagger}A\|_{\xi}$, figures 3.1, 3.2, 3.3, and Table 3.4 show that our technique (WS) is best at minimizing the residual. We further analyze the selected features and their ranking on the Iris, CorAl, Madelon and S&P datasets.

The Fisher Iris dataset has four features, two of which, namely the 3^{rd} and the 4^{th} are good enough to distinguish the underlying classes [66, 112]. As can be seen in Table 3.2, four of the compared techniques (including ours) accurately picked the 3^{rd} and the 4^{th} features as the most relevant features. Our unsupervised Weighted Score (WS) technique and the supervised Fisher Score(FS) technique provided the

Table 3.2: Iris & CorAl Features Ranking

Techniques	Feature number				Feature number				
WS	3	4	1	2	1	2	3	4	6
FSPCA	1	4	2	3	2	3	6	5	1
LS	4	3	2	1	1	3	4	2	6
LSS	4	3	2	1	1	4	6	3	2
Clever	4	1	2	3	2	4	5	6	3
FS	3	4	1	2	1	2	3	4	6
RfF	4	3	1	2	1	3	2	4	6
mrmr	4	2	3	1	1	6	2	3	4
Datasets	**(Fisher Iris)**				**(CorrAl)**				

same ranking whereby we have features 3, 4, 1 then 2.

The CorAl dataset is known to have 4 relevant features{1, 2, 3, 4}, one irrelevant feature {5} and one highly correlated {6} with a 25% error rate (matches the class label 75% of the time) [66]. All compared methods except for FSPCA [90] and the Leverage Score technique [42] consistently rank the 4 relevant features {1, 2, 3, 4} above 6 and 5 (see Table 3.2).

On the Madelon dataset, our proposed technique (WS) along with ReliefF successfully returned all the 20 relevant features. Although, as seen in Table 3.5, the different methods however return a different ranking of the 20 features. Fisher Score and mrmr also return a good number of the 20 relevant features, respectively 13 and 12 (Table 3.5). Consequently Fig. 3.2, and Table 3.5 show that with 20 features retained out of the original 500, those four techniques are better at minimizing the residual $\|A - CC^{\dagger}A\|_{\xi}$ than the other techniques.

We further assess the performance of the various techniques with different sizes of retained features. Fig. 3.4 shows the reconstruction error $\|A - CC^{\dagger}A\|_{\xi}$ for a number of selected features between 5 and 35. Fisher Score, mrmr, ReliefF, and WS exhibit a sharp decrease of the residual for selections between 5 features and 20 features, and a more controlled decrease after 20 features. This behavior illustrates the fact that,

Tech.	Feature number				
WS	15	19	17	21	13
FSPCA	19	25	27	5	14
LS	14	20	22	24	19
LSS	11	6	18	19	31
Clever	8	14	17	22	31
FS	5	3	7	1	31
RfF	8	6	16	24	14
mrmr	2	5	3	1	7
Dataset	Ionosphere				

Table 3.3: Ionosphere top 5 features selected by different techniques

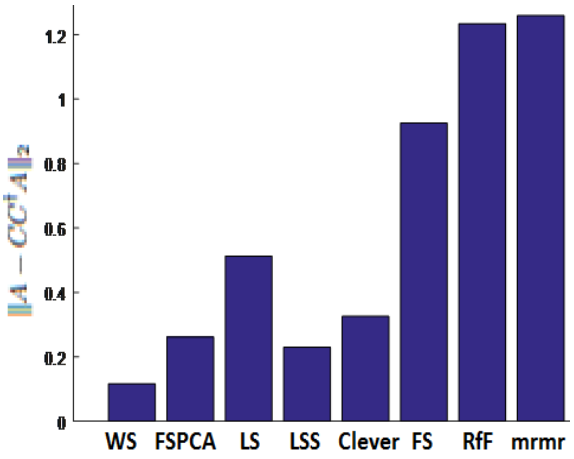

Figure 3.1: Inosphere residual minimisation for 5 features

Table 3.4: Minimizing the Residual $\|A - CC^{\dagger}A\|_{\xi}$ on Different Datasets Using Different Techniques

	Arrethmia (20 of 34)		Gisette (20 of 5000)		Ionosphere (5 of 34)		Madelon (20 of 500)		Mallat (15 of 1025)		Reuters(20 of 5180)		S&P(20 of 427)	
Techniques	Residual	Time	Residual	Time	Residual	Time	Residual	Time	Residual	Time	Residual	Time	Residual	Time
WS	1.657	0.031	376.836	1.374	0.116	0.004	0.010	0.100	0.022	0.979	0.002	6.122	0.0187	0.069
FSPCA	4.585	0.021	3257.888	1.360	0.262	0.005	175.060	0.100	0.031	0.986	3.058	6.139	0.797	0.073
LS	7.574	0.021	3259.343	1.380	0.513	0.004	104.050	0.100	0.118	1.018	1.554	6.522	0.306	0.070
LSS	5.619	0.024	1369.852	1.346	0.335	0.009	17.740	0.100	0.096	1.003	3.376	6.309	0.396	0.068
Clever	32.814	0.732	4181.725	2017.580	0.326	0.058	5.670	4.500	0.091	3.399	3.565	431.475	0.286	0.546
FS	5.665	0.086	2726.849	0.142	0.925	0.004	0.080	0.100	0.029	0.271	1.788	455.186	1.196	0.038
RfF	14.033	0.853	669.300	43.737	1.234	0.177	0.010	15.700	1.098	66.763	0.542	112.379	0.252	5.570
mrmr	52.861	0.087	4181.725	0.598	1.259	0.007	0.080	0.400	0.774	1.045	5.020	0.925	1.112	0.358

for up to 20 features, those particular techniques are progressively selecting relevant features with higher weights within the dataset. These observations are consistent with what is known of the dataset (it was generated with 20 relevant and 480 noise features). As seen in Fig. 3.4, while the four techniques(mrmr, Fisher Score, ReliefF and WS) select some of the same features for subset sizes between 5 and 35, WS appears to be selecting the best set each time, followed by ReliefF and mrmr. In particular ReliefF and WS select the same set of 20 best features. We must note that, of the four feature selection techniques performing well on this dataset, three techniques (mrmr, Fisher, ReliefF) are supervised, while our technique (WS) is unsupervised. Experiment results on the time series dataset S&P shown in Fig. 3.5 indicate that our technique consistently outperforms the remaining techniques for selected features

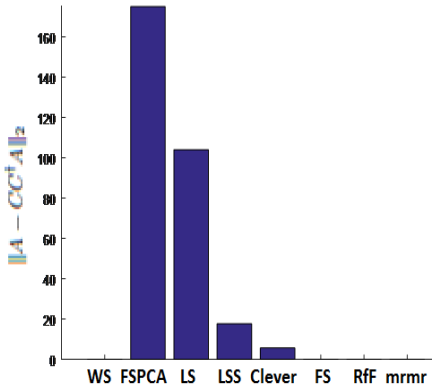

Figure 3.2: Madelon residuals minimization for 20 features

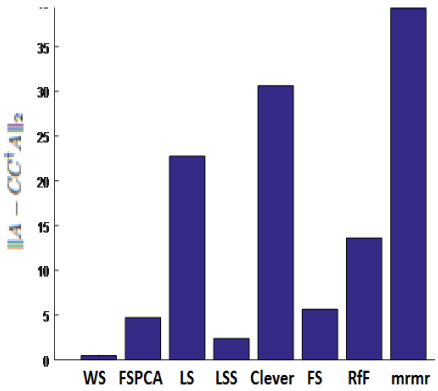

Figure 3.3: Arrethmia residual minimization for 5 features

Table 3.5: Madelon Dataset 20 Most Representative Features Selected Using Different Algorithms

Techniques	20 Most Relevant Features Selected from Madelon																			
WS	106	337	456	65	494	339	454	154	476	434	319	242	282	379	129	49	443	29	473	452
FSPCA	*223*	*256*	*407*	*251*	*331*	*18*	*148*	*322*	*188*	*221*	*406*	*200*	*413*	*243*	*313*	*52*	*402*	*334*	*350*	*286*
LS	*69*	*388*	*41*	*138*	*224*	*4*	64	*74*	*1*	*86*	*141*	*170*	*171*	175	*264*	*448*	*452*	*15*	*17*	*37*
LSS	*268*	*401*	*218*	*179*	*106*	*445*	*23*	*162*	*74*	*301*	*463*	*204*	*486*	*209*	*479*	*227*	*40*	*361*	*99*	*369*
Clever	29	106	282	339	*54*	*77*	*100*	107	*139*	*196*	*237*	*246*	*275*	*285*	*359*	*406*	*442*	*467*	*467*	*482*
FS	476	242	337	65	129	106	49	379	339	443	473	454	494	*324*	*425*	*206*	*412*	*205*	*283*	*297*
RfF	379	49	476	242	319	339	443	29	473	452	494	154	282	434	454	106	337	129	65	456
mrmr	242	49	129	443	337	379	454	65	473	494	339	106	*297*	*324*	*11*	*425*	*476*	*299*	*283*	*412*

between 20 and 300.

3.4.3.2 Discriminative Power of the Selected Features

In this set of experiments we investigate how well do the subsets of features, selected by different algorithms, help in separating classes. For that purpose, different feature selection algorithms are used to identify and select the top-k features from each dataset (20 features in both cases of the Synthetical- Control dataset in Fig. 3.6 and the Soybean dataset in Fig. 3.7).

The synthetical-Control dataset consists of synthetically generated control charts data to help define the notion of similarity between time series. In some cases the Euclidean distance can be quite large while time series are still similar. Hence Alcock

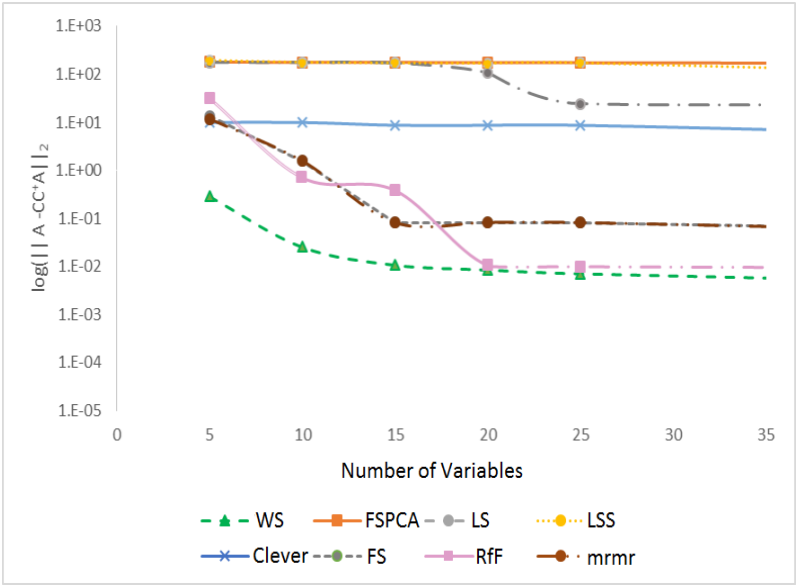

Figure 3.4: Reconstruction error $\|A - CC^{\dagger}A\|_{\xi}$ for selected features (5 to 35) on the Madelon dataset.

and Manolopoulos [1] proposed to transform the series using the DFT, into feature vectors prior to computing their similarities using a distance function. The data is stored in an ASCII file, with 600 rows and 60 columns, a single chart per line, and 6 classes (Normal, Cyclic, Increasing trend, Decreasing trend, Upward shift and Downward shift). Because the Fourier space preserves the Euclidien distance of two signals [1], the dataset provides a good testing space for clustering and classification problems.

The Soybean dataset was gathered from information regarding plant conditions(e.g. mold growth) and environmental conditions such as temperature, precipitation etc. to help predict diseases in soybean crops. In this experiment, the small version of the dataset was used with four classes of diseases(diaporthe-stem-canker(D1), Charcoal Rot(D2), Rhizoctonia Root Rot(D3), Phyrophtora Rot(D4)).

The selected subset of features for each dataset(Synthetical-Control Charts and Soybean) is used to perform the PCA and original data samples are projected onto the first two principal components.The classes within both datasets each present a

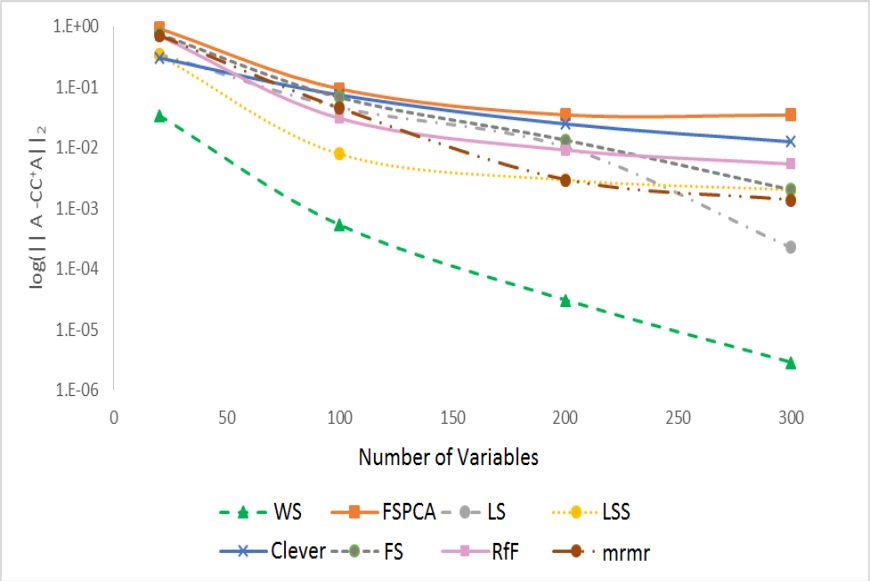

Figure 3.5: Reconstruction error $\|A - CC^{\dagger}A\|_{\xi}$ for selected features(20 to 300) on the S&P dataset.

number of dots which all bear the same color as seen on Fig. 3.6 and Fig. 3.7. The results on Fig. 3.6 show that, along with the Fisher Score technique, WS is best at separating the classes in the Synthetical-Control Charts dataset. Furthermore, the results in Fig. 3.7 shows that our technique outperforms all the remaining techniques in classifying the Soybean dataset.

3.4.3.3 Classification Improvement with Feature Elimination

Our feature selection technique was also evaluated on the MLSP 2014 Schizophrenia Classification Challenge dataset [63]. In this specific experiment we looked to assess how the proposed feature selection technique contributed to improving classification. We used two classifiers from the Matlab Statistics and Machine Learning Toolbox: the linear Super Vector Machine(SVM) and Subspace K-Nearest-Neighbors(KNN). In both cases, the classification accuracy improves as we eliminated 45% of the features.

Our technique shows that the first principal component carried 99.54 of the explained variance (Table 3.6), and returns 55 percent of the features (101 features) as

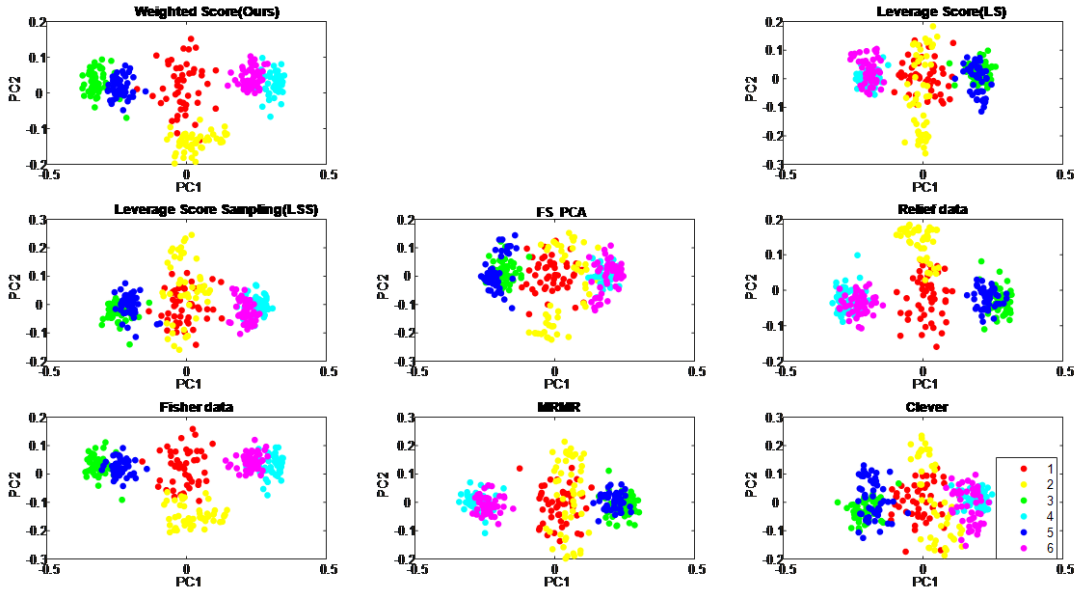

Figure 3.6: Synthetic Control data projection on the first two principal components

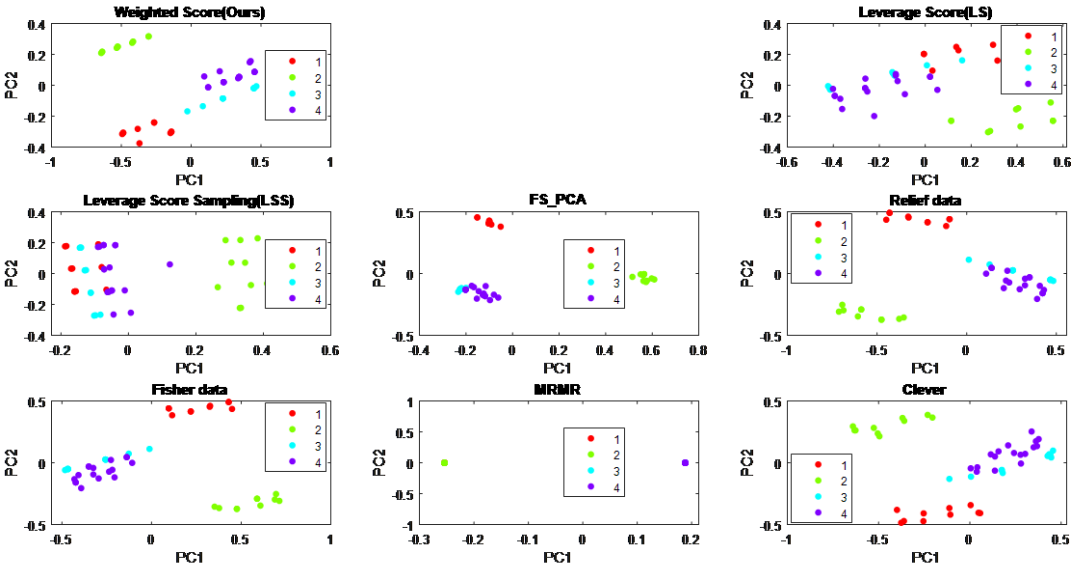

Figure 3.7: Soybean data projection on the first two principal components

Table 3.6: Percentage of variance explained for the first 6 principal components

Principal Component	PC1	PC2	PC3	PC4	PC5	PC6
Variance Explained (%)	99.545	0.226	0.142	0.05	0.02	0.017

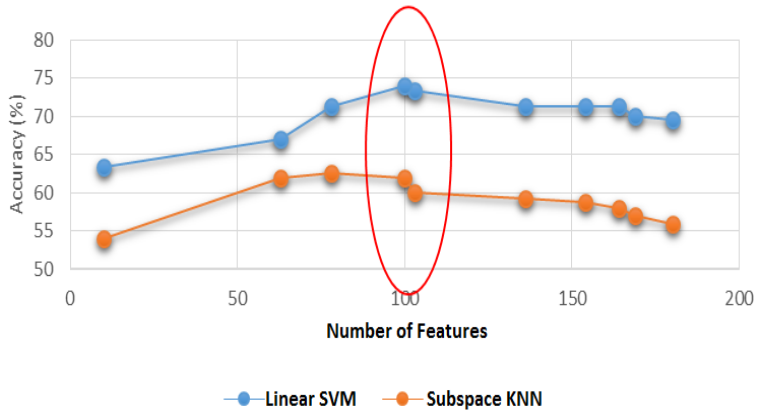

Figure 3.8: Classification improvement with feature elimination

relevant (Fig. 3.8).

On one hand when the number of retained features is below 101, the accuracy decreases, due to the fact that relevant features are missing. On the other hand, when the number of retained features is over 101, the accuracy is affected due to the fact that irrelevant features were being introduced.

Identifying and leveraging the relevant set of features allows improving classification accuracy in the this dataset.

3.5 Summary

In this chapter we proposed a new feature selection technique using Principal Component Analysis. It leverages the desired properties of the principal components by identifying the features that allow to retain the maximum variability of the data, hence to minimize the reconstruction error. Our experiments on numerous real datasets indicate that while our technique picks the top most representative k features in terms

of accuracy, its computation time is comparable to the other efficient PCA based techniques while it enjoys at least an order of magnitude better than other techniques.

Chapter 4: Feature Selection, Grouping and Engineering

We also study the problem of uncovering the most relevant and discriminative features in sparse MTS and in environments where dependent features work better together. Existing solution techniques often rely entirely on feature selection. We believe that one should also consider interactions among the feature vectors and combine feature section to feature grouping for improved results. In this chapter, we propose an unsupervised feature selection and grouping technique that reduces noise, identifies relevant features and groups correlated ones. For this, we first apply unsupervised learning through a randomized PCA to uncover influencial features and rank them accordingly. The correlated features are then identified, grouped, and recombined into unique feature vectors to allow scalable and high performance query processing over high dimensional MTS. We carried out numerous experiments to evaluate the performance of the proposed technique using well-known benchmark datasets in different application domains. Our results indicate improved efficiency while providing increased accuracy in most cases.

In what follows, we review the background and preliminaries in Section 4.1, discuss the related work in Section 4.2, and introduce the technique in Section 4.3. Section 4.4 presents the performance evaluation. A summary, our concluding remarks and future directions are presented in Section 4.5.

4.1 Background and Preliminaries

In many fields, practitioners work with large amounts of MTS data to find interesting and meaningful patterns. Effective management and processing of such data have progressively become very challenging due to the overwhelming growth in volume, high dimensionality and complexity, as well as increased amount of noise and redundancies in such frameworks. Processing such data often leads to high computational cost and massive memory requirements. Furthermore, identifying and extracting useful information require preprocessing steps such as feature grouping or dimensionality reduction for many practical applications. Dimensionality reduction can often be achieved through feature selection or extraction, both of which yield a rich reduced set of features, however they differ in their approaches.

On one hand, feature extraction relies on transforming the existing feature variables and creating a new and often reduced number of richer features as a function of the original features. For instance, in the particular case of PCA, the new features are conceived so to have the largest possible variance, since the percentage of explained variance retained in the new variables indicates the amount of information retained within the reduced data. The larger the variance retained, the lower will be the distortion at reconstruction. Feature extraction techniques often can uncover a new optimal embedding space in linear time, making them preferred in terms of computational complexity. They are generally based on preprocessing steps such as normalization, standardization, discretization, signal enhancement, coordinates transformations or local feature extraction. Unfortunately, such transformation processes can render the newly extracted features difficult to interpret.

On the other hand, feature selection (also known as variable or attribute subset selection) identifies the subset of features that best captures adequate information from the original set to be used for successful and enhanced subsequent processes. The immediate interpretability that results from retaining a subset of the original variables makes feature selection as a preferred dimensionality reduction strategy for practitioners in cases where the construability of the acquired subset of features with

respect to the original features is to be maintained, or when the number of features is sizeably larger than the number of samples [27]. The feature selection process is however a combinatorial optimization problem, which is NP-Hard. To overcome that challenge, most proposed techniques rely on solving, learning, or discovery methods to uncover the set of relevant features in reasonable time.

While feature subset selection aims to identify and retain a reduced set of the most influential features within a dataset, the existing techniques tend to overlook the underlying relationships between the feature vectors. Investigating those relationships often reveals the presence of groups within the given data which leads to uncovering important characteristics that could otherwise be missed. The importance of such groupings particularly reveals itself in sparse feature frameworks or in cases where dependent features are known to work better together in groups than individually. Recent research directions, have for instance, shown that many datasets in practice present a structure such that correlated features work better in groups than when considered individually [110]. Hence feature dependence or correlation has become one of the most widely used means of uncovering groups within a dataset. We present a simple, yet efficient unsupervised feature selection technique that reduces noise, identifies the most influential features and groups the correlated ones.

The technique first relies on unsupervised learning, hence data driven discovery, through randomized PCA to uncover influence and rank features accordingly. Correlated features are then subsequently identified, grouped and, re-structured into unique features to allow for a more efficient and scalable processing of high dimensional MTS.

The technique combines advantages from both feature extraction and selection techniques by leveraging desired properties from the PCA, and retaining interpretability through the selection of a subset from the original features. It also reveals more insights about the data and improves accuracy by uncovering and leveraging feature groups within the given subset. Our experimental results on a large number of application domains and well studied benchmark datasets indicate its performance.

4.1.1 Problem Formulation

A MTS $A_{n,m}$ of n instances with m variables can be represented as an $n \times m$ matrix A (shown below) in which $t_{i,j}$ is the value of variable v_j measured at time-stamp i, for all $1 \leq i \leq n$ and $1 \leq j \leq m$.

$$A_{n,m} = \begin{bmatrix} t_{1,1} & t_{1,2} & \cdots & t_{1,m} \\ t_{2,1} & t_{2,2} & \cdots & t_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ t_{n,1} & t_{n,2} & \cdots & t_{n,m} \end{bmatrix}$$

We are interested in the problem of unsupervised feature selection and grouping that can be framed from a general perspective as follows:

Let A be a data matrix of n instances and m features. The goal is to find the most "informative" features within the data matrix, by data driven discovery, hence without relying on class labels to guide the search. By "informative" we mean features that best capture the underlying structure of the data. We also seek to group and recombine correlated features within the selected subset. The expectation is to preserve as much as possible of the original data intrinsic structure within the selected subset of relevant and non redundant features, to allow for enhanced results when conducting downstream processes.

The unsupervised feature subset selection problem is often considered an instance of the column subset problem (CSS) [13, 10] defined in Section 3.2.

We use a randomized SVD algorithm [30], to identify and retain the k first principal components according to the relative percentage variance criterion. The k principal components carry the amount of variance we wish to retain from the data.

4.2 Related Work

The literature pertaining to feature selection in general is vast and can be placed back to about six decades ago [6, 8]. Early techniques where often based on probabilistic strategies [37]. A large variety of strategies have since been proposed. Recent

literature is increasingly directed towards strategies looking to address big data challenges pertaining to the high volume and dimensionality, as well as to the increasing amounts of noise and redundancies.

More specifically, and although scarce, some effort has been made to investigate feature selection in MTS [108, 32, 16, 106]. Those techniques are unfortunately often supervised which can be expensive in large dataset frameworks. There is a need to develop techniques better suited for time series data in such frameworks.
While the proposed feature selection techniques generally succeed in identifying a subset of most relevant features, they often fail to uncover and leverage some important relationship between variables, which would otherwise provide more insight into the data. Consequently, feature grouping is increasingly becoming a very important tool to further reduce the dimensionality, bring out the intrinsic relationship that could otherwise be missed and eliminate redundancies [113, 110, 96]. This is particularly true in sparse vector environments where feature grouping yields very good results [110]. Leveraging feature interactions is also important in practical applications such as in Bio-informatics where dependent features are known to work better in groups than on their own. For instance studies have shown that, due to their functionality, genes are required to be considered within given groups for more meaningful results, when gene activity studies are being conducted [65].

Early feature grouping techniques include techniques from the fused lasso family such as fused Lasso [95], graph based fused lasso [59] and a generalization of the fused Lasso technique [25], but also Elastic-Net [115], OSCAR (octagonal shrinkage and clustering algorithm for regression) [7] and, the Variable Grouping in Multivariate Time Series Via Correlation [96]. The fused lasso family of techniques uses some sparse fused regularizers to penalizes the differences between coefficients and accordingly link features. These techniques however are often not able to find the feature groups automatically from the data but rather require them to be provided in order to achieve sparse modeling. The Elastic-Net [115] technique is considered an extension of the lasso technique and provides similar sparsity representations. It uses both ℓ_1 and ℓ_2 regularizers and forces the coefficients of highly correlated features to be close

in high dimensional frameworks. While this is very efficient in grouping highly correlated features, it can miss cases where variable are highly correlated but of different magnitudes. OSCAR [7] allows for feature selection and automatic feature grouping using ℓ_1 and ℓ_2 regularizer and a pairwise ℓ_∞ regularizer; the ℓ_1 regularizer is used for feature selection purposes, while the ℓ_∞ regularizer serves to group the features and reduce redundancies automatically. It is however not always easily applicable due to the complex nature of its proximity operator and its optimization process that can be computationally expensive. The technique proposed in [96] groups MTS by primarily leverages time lagged correlations between variables to uncover relationships. The MTS is decomposed into smaller groups of MTS, where variables within the same group are highly correlated according to Spearman correlation measure, while they are relatively independent of one another when residing in different groups. The technique uses a genetic algorithm among other strategies for grouping features.

Much of the recent effort for feature grouping has also relied on feature dependence measurements, such as Mutual Information(MI) [91, 62] or correlation coefficient measures [68, 110] to study the underlying structure of the data. The mutual information is often used in these settings when the goal is to quantify the amount of information that one can gain about one variable from what is known about another variable, while the correlation coefficient is rather used to reveal the strength of the linear dependencies between two variables. While some progress has been noted in recent years, more work pertaining to feature selection and grouping is needed to better address today´s data challenges in high dimensional and high volume frameworks.

4.3 FRG: Feature Ranking and Grouping

The proposed technique, FRG presents a two steps approach. In the first step, it uncovers the feature relevance and ranks the variables accordingly. It subsequently identifies, groups, and combines correlated features in the second step.

Given a MTS represented in the form of a matrix $A_{n\times m}$ and a correlation threshold ϵ, our goal is to identify the most influential features (according the their weights)

while grouping and recombining features that are correlated to those influential features, within the chosen subset, with a correlation value greater than ϵ.

Definition 4.1. *(Relevant and Primary) Let $A \in R^{n \times m}$ be a matrix with rank $r = rank(A)$ such that $r \leqslant min\{n, m\}$ and $k \leqslant r$. Let V_k be the matrix containing the top k right singular vectors of A, and S_k be the matrix containing the top k singular values of A. We consider D^U, the set of m features (UTS) that make up A, and $D^{U(k)}$ (sorted in descending order) the subset of the k most relevant features from D^U. Then, we say a feature X_i of the matrix A is* ***"Relevant and Primary"*** *if, for all $X_i, X_j \in D^{U(k)}$ such that $i < j$, whenever $\rho(X_i, X_j) \geq \epsilon$, then $\hat{w}_i^{(k)} \geq \hat{w}_j^{(k)}$. X_j is subsequently included in the group G_{X_i} containing X_i and removed from $D^{U(k)}$, with:*

- *$D^{U(k)}$ is the set of the k most relevant features of A.*
- *G_{X_i} is the group containing all features correlated (with correlation over ϵ) to X_i, and X_i is the primary feature in that group.*
- *$\rho(X_i, X_j)$ is the pairwise Pearson correlation value of X_i and X_j, and ϵ the correlation threshold*
- *$\hat{w}_i^{(k)} = |\Sigma_{j=1}^{k} w_j v_{i,j}|$, for $1 = 1, 2, ..., m$,*
  - *$\hat{w}_i^{(k)}$ is the i^{th} element entry of $\hat{w}$, and the weight of feature variable X_i.*
  - *$w_j = \lambda_j / \Sigma_{z=1}^{r} \lambda_z$, the fraction of variance carried by the j^{th} column in V_k, for $j = 1, 2, ..., k$*
  - *$\lambda_j = \sigma_j^2/(n-1)$ is the variance corresponding to the j^{th} singular value, consequently to the j^{th} column of V_k, and $\lambda_1 \geq \ldots \geq \lambda_r \geq 0$*

4.3.1 Feature Weighting and Ranking

We use randomized PCA to reveal the relevance of the given features and rank them accordingly. We expect this to help chose the subset of features that best captures the structure of the original data. Let us consider the factorization A $\approx U_k S_k V_k^T$ resulting

from the SVD of the matrix A. From this factorization, our proposed technique uses matrix S_k which contains the singular values, and matrix ${V_k}^T$, whose rows represent the eigenvectors of the covariance matrix $A^T A$ to compute statistics that will reveal relevance. The basic idea that allows to compute those statistics known as variable weights or weighted scores [48] stems from the composition of the factoring matrices S_k and V_k. Indeed, the column entries of a vector within V_k provide the regression coefficients of its corresponding principal component, which in turn is expressed as a linear combination of all the variables from the original matrix. The most relevant variables for the given principal component are reflected through the largest positive or negative coefficients of the linear combinations. Hence, the entries within each eigenvector already provide a sense of how important or influential each variable is within the specific principal component.

In addition when considered with respect to the whole dataset, each eigenvector reflects a different level of importance. Indeed, they represent in decreasing order an explained amount of variance from the data. The diagonal entries of the matrix S_k, also sorted in decreasing order of importance, allow to uncover the fraction of variance carried by each principal component. For instance, with σ_j as the j^{th} diagonal entry of S_k, the fraction of explained variance retained by the specific j^{th} principal component is computed as $w_j = \lambda_j / \Sigma_{z=1}^{r} \lambda_z$ with $\lambda_j = \sigma_j^2/(n-1)$.

Let us consider the weight vector $\hat{w}$ containing all the m variable weights. To obtain the weight $\hat{w}$, we multiply each principal direction by its importance, or corresponding percentage of variance. Hence, we multiply the matrix V_k of principal directions by the vector Λ_k carrying the respective fractions of explained variances for the principal directions. The weights' vector $\hat{w}$ is then expressed as $|V_K \Lambda_k|$ and, its i^{th} entry $\hat{w}_i^{(k)}$ corresponds to the weight of the i^{th} original variable X_i expressed as $\hat{w}_i^{(k)} = |\Sigma_{j=1}^{k} w_j v_{i,j}|$.

The variables are subsequently sorted according to their weights in decreasing order to reveal the most influential ones. To identify the number k of variables to retain we use the Broken-Stick method [64]. Steps 1 to 4 in the Algorithm summarize the variable weighting and ranking process.

4.3.2 Feature Grouping and Reduction

The second step in our proposed technique consists of identifying dependent features, gathering them into groups and re-combining each group into a univariate feature through PCA.

We use Pearson's product-moment coefficient [76] as the measure to assess the dependence between features. The Pearson correlation measure is known to be robust against data that is not normalized and to respond better to baseline and scale shifts when compared to other measures [114].

Let X_i and X_j be two features from A. The Pearson correlation coefficient of X_i, X_j denoted $\rho(X_i, X_j)$ is a value in [-1,1] that measures the linear dependency between X_i and X_j , defined as follows:

$$\rho(X_i, X_j) = \frac{\sum_{t=1}^{n} (x_i - \overline{x_i})(x_j - \overline{x_j})}{\sqrt{\sum_{t=1}^{n} (x_i - \overline{x_i})^2}\sqrt{\sum_{t=1}^{n} (x_j - \overline{x_j})^2}} \tag{4.1}$$

where $\overline{x}_i$ is the mean of X_i over n and $\overline{x_j}$ is the mean of X_j over n. The Pearson correlation coefficient can be approximated to the Pearson product moment, expressed as follows:

$$\rho(X_i, X_j) = \frac{1}{n-1} \sum_{t=1}^{n} \frac{x_i x_j}{S_{xi} S_{xj}} \tag{4.2}$$

where $x_i = (x_i - \overline{x_i})$, $x_j = (x_j - \overline{x_j})$, $S_{xi} = [(1/n-1) \sum_{t=1}^{n} {x_i}^2]^{1/2}$,
and $S_{xj} = [(1/n-1) \sum_{t=1}^{n} {x_j}^2]^{1/2}$.

As defined in Definition 4.1, a feature X_i of the data matrix A is considered *relevant* and *primary* if for every X_j in the sorted set $D^{U(k)}$ of the k most relevant features of A, whenever $\rho(X_i, X_j) \geq \epsilon$, it holds that $\hat{w}_i^{(k)} \geq \hat{w}_j^{(k)}$ and that X_j is included in the same group G_{X_i} that contains X_i, and hence removed from $D^{U(k)}$. The feature grouping and reduction process happens as follows:
Given a user specified correlation threshold ϵ, our goal is to identify the most influential features and group them with lesser influential ones that are strongly correlated to them (with correlation value greater than the given threshold ϵ), within the selected

subset. The approach unfolds according to the following steps:

1. Pick the first element X_i from the ordered set $D^{U(k)}$ of k most influential features (sorted in descending order).

2. Group correlated features:

   - Find every other feature in $D^{U(k)}$ whose correlation with X_i is not less than a threshold value ϵ. Remove all such features from $D^{U(k)}$ and include them into a group called G_{X_i} in which X_i is the primary feature. If no such feature exists, then X_i will be the only feature in G_{X_i}; keep X_i and the process continues at step 4.

3. Run a local randomized PCA [30] on the group G_{X_i} and select the first principal component's scores as the new feature to represent the whole group.

4. Pick the next element of the updated ordered set of features. Steps 2 and 3 are repeated until no more feature could be added to the list of most influential and non redundant features.

Algorithm 4.1 summarizes the steps for feature grouping and reduction from line 6 to 15.

4.4 Experimental Set Up and Results

To evaluate the effectiveness of our proposed selection and grouping technique FRG, we implemented the proposed algorithm in Matlab and conducted numerous experiments on benchmark datasets, using a configured PC with Intel Quad core i7 2.00 GHz CPU, 8 GB RAM, and running Windows 7.

4.4.1 Benchmark Datasets

The experiments were ran on over fifty synthetical and real time series benchmark datasets, from a wide range of domain applications, including UCR time series classification archive [55], UCI repository [66], FMRI datasets from [44, 63, 19], and

Algorithm 4.1 - Feature Ranking and Grouping (FRG)

Input: $A \in R^{nxm}$

Output: $S_r \in R^{n \times k}$ is spawned by the k most influencial features of A.

begin

FEATURE RANKING

Compute the truncated Singular Value Decomposition
$[U_k, S_k, V_k^T] \leftarrow Randomized\_SVD(A)$

Uncover the variable weights and populate $\hat{w}^k$, the vector of weighted scores
$\hat{w}^k \leftarrow |V_K \Lambda_k|$ with $\Lambda_k = diag(S_k^2./\Sigma_{i=1}^k s_{ii})$
Note: $\hat{w_i^k}$ corresponds to the weight of the i^{th} feature variable and can be expressed as $\hat{w}_i^k = |\Sigma_{j=1}^k \lambda_j v_{i,j}|$

Sort the variables according to their weights in decreasing order both in the weights vector $\hat{w}$ and the set of features D^U
$\hat{w}^k \leftarrow \hat{w}_1^k \geq ... \ \hat{w}_i^k \geq ... \geq \hat{w}_m^k$
$D^U \leftarrow X_1 \geq ... \ X_i \geq ... \geq X_m$

Uncover the number k of most influential features to retain
k $\leftarrow$ Broken_stick_method($\hat{w}^k$)
$D^{U(k)} \leftarrow X_1 \geq ... \ X_i \geq ... \geq X_k$

FEATURE GROUPING

i $\leftarrow$ 1,

Set the first element of the sorted set $D^{U(k)}$ to be X_i

while($exist(GetNextFeature(D^{U(k)}, i))$ **) do**
$X_i \leftarrow SetNextFeature(D^{U(k)}, i)$
$j \leftarrow i$

Successively set the remaining items from $D^{U(k)}$ to X_j to assess how dependant they are with X_i

while($exist(GetNextFeature(D^{U(k)}, j))$ **) do**
$(X_j \leftarrow SetNextFeature(D^{U(k)}, j))$
$j \leftarrow j + 1$

Compute their pairwise correlations
$\rho_{ij} \leftarrow |\rho(X^i, X^j)|$

Insert the features correlated to X_i in the feature group G_{X_i} and delete it from the set $D^{U(k)}$
If $(\rho_{ij} \geqslant \varepsilon)$ then add X_j to set G_{X_i} and delete X_j from $D^{U(k)}$

end while

If G_{X_i} is not empty then add X_i to G_{X_i} and compute the local Randomized PCA of the Group G_{X_i}

Select the first PC's scores to replace the feature X_i, at its original position in $D^{U(k)}$
$i \leftarrow i + 1$

end while

return S_r

end

Table 4.1: Benchmark Datasets

Datasets	Features	Instances	Classes
CBF	900	128	3
Coffee	28	286	2
Coil	17	340	2
Congress	16	435	2
ChlorineC	3840	166	3
CorAl	6	32	2
EEG Arethmia	279	452	16
ECG 5000	4500	140	5
EEG EyeState	15	14980	2
Face	350	88	4
FMRIStarplus	80	74000	2
Heart	13	270	2
Madelon	500	1800	2
Gisette	5000	6500	2
Ionesphere	34	351	2
Iris	4	150	3
Pima	8	768	2
Reuters	5080	1806	2
Soybean	35	47	4
Synth_ctrl	60	600	6
Trace	100	275	4
Two Patterns	4000	128	4
UPS	13	270	2

financial market datasets [78]. In an attempt to uncover how the proposed technique would fair against other techniques on classical well-known datasets specifically designed for feature selection problems, we also conducted experiments on datasets other than time series. For these we used 5 high-dimensional data sets from [27], 25 small benchmark datasets from [26], and 12 small datasets used in [11]. Our aim was to be as thorough as possible by considering a wide variety of datasets with different and challenging characteristics for feature selection. We provide more details about the datasets discussed in this section in Table 4.1 and in what follows.

4.4.2 Peer Techniques

We compared our results against those of the following eight supervised and unsupervised techniques:

- **Clever** [105] is unsupervised, using PCA and leveraging a Common Principal Components (CPCA) between MTS.
- **FSPCA** [90], feature selection using principal component analysis (FSPCA) is unsupervised. It exploits results from a PCA of the covariance matrix to evaluate the significance of each feature component.
- **Fisher Score(FS)** [20] is a supervised technique that seeks a subset of features, such that, the distances between data points reflect memberships to classes.
- **Leverage Score(LS)** [42] is unsupervised and,relies on PCA. It samples features from the original matrix that correspond to the largest leverage scores.
- **Leverage Score Sampling(LSS)** [18] is unsupervised, considered a randomized version of the technique proposed in [42]. It samples features corresponding to probabilities proportional to its largest leverage scores.
- **Max-Relevance Min-Redundancy(mrmr)** [77] is supervised and selects good features according to the maximal dependency criterion based on mutual information.
- **ReliefF(RfF)** [61] is supervised and estimates the quality of attributes according to how well their values distinguish between instances that are near to each other.
- **Weighted Scores(WS)** [48] is an unsupervised learning technique that identifies the top-k discriminative features by leveraging statistics drawn from the principal components. Its intuition is based on the fact that principal components must have different weights when being recombined in a new framework with respect the whole dataset. Our proposed feature selection and grouping technique(FRG) is an extension of the Weighted Score (WS) technique.

Our proposed technique leverages variance as a criterion to identify and rank relevant features and uses Pearson correlation as a way to assess the dependence between the selected features. Other techniques based on variance that we considered and compared to our technique include Fisher Score [20], Leverage Score [42, 18], Clever [105], and FSPCA [90]. In these techniques either the highest ranked features are chosen [42, 105, 20, 90] or features are randomly sampled with a probability proportional to their importance [18]. We also compared our results with techniques based on other criteria such as dependence (e.g., Max-Relevance Min-Redundancy(mrmr) [77]). Our proposed technique outperforms the other techniques in terms of speed and accuracy on a large number of datasets as shown in our results, details of which follow.

4.4.3 Evaluation and Results

We designed sets of experiments described as follows.

4.4.3.1 Ranking Features and Minimizing the Residual $\|A - CC^{\dagger}A\|_{\xi}$

For this set of experiments, we evaluate the performance of our Feature Selection and Grouping Technique (FRG) against other techniques on a large number of datasets. For each dataset, we retain the set of k most relevant features and subsequently evaluate the different methods according to their ability to minimize the residual $\|A - CC^{\dagger}A\|_{\xi}$. Here, A is the original data matrix with all features, C is the matrix built out of the k identified features, and $\xi = \{2, F\}$, where the 2 denotes the Euclidean norm and F denotes the Forbenium norm. The goal is to identify which technique yields the best set of features.

In minimizing the residual $\|A - CC^{\dagger}A\|_{\xi}$, figure 4.1 to 4.3 show that our techniques (FRG and WS) are best at minimizing the residual. We must note that, the first step of our proposed technique (FRG) provides the same set of features than the Weighted Scores (WS) technique. In cases where the selected features are not correlated over a given threshold ϵ, the selected subsets of features remain the same. However, for cases where linear dependencies are noted between features, with correlations values greater than ϵ, the correlated features are re-grouped and recombined, to provide

Table 4.2: Iris & CorAl Features Ranking

Techniques	F. number				F. number				
FRG	3	4	1	2	1	2	3	4	6
WS	3	4	1	2	1	2	3	4	6
FSPCA	1	4	2	3	2	3	6	5	1
LS	4	3	2	1	1	3	4	2	6
LSS	4	3	2	1	1	4	6	3	2
Clever	4	1	2	3	2	4	5	6	3
FS	3	4	1	2	1	2	3	4	6
RfF	4	3	1	2	1	3	2	4	6
mrmr	4	2	3	1	1	6	2	3	4
Datasets	**(Iris)**				**(CorrAl)**				

Table 4.3: Ionosphere top 5 features selected by different techniques

Tech.	Feature number				
FRG	15	19	17	21	13
WS	15	19	17	21	13
FSPCA	19	25	27	5	14
LS	14	20	22	24	19
LSS	11	6	18	19	31
Clever	8	14	17	22	31
FS	5	3	7	1	31
RfF	8	6	16	24	14
mrmr	2	5	3	1	7
Dataset	**Ionosphere**				

one feature per group. The Madelon [27, 66] dataset presents an example of such cases. It is made of Gaussian clusters positioned on the vertices of a hypercube and labeled randomly. The dataset was generated with 20 relevant and 480 noise features and considered high dimensional. On this dataset, our proposed technique (FRG), in its first step, along with Weighted Scores (WS) and ReliefF successfully returns all 20 relevant features. The correlated features selected by FRG are subsequently regrouped and re-engineered in its second step. Table 4.4 shows the resulting set of ten features for FRG, when the correlation threshold ϵ is set to 0.98. Fisher Score and mrmr also return a good number of the twenty relevant features, respectively 13 and 12 (see Table 4.4). Consequently Fig. 4.1 shows that, with 20 features retained out of the original 500, those five techniques are better at minimizing the residual $\|A - CC^{\dagger}A\|_{\xi}$ than the remaining techniques.

Feature ranking is also an important aspect in both WS and FRG. It uncovers early in the process which features best capture the structure of the data and, specifically for FRG, it helps identify the features that are *relevant and primary*. We further investigate the performance of the various techniques on the Madelon, Iris and CorAl datasets in relation with feature ranking. Fig. 4.3 shows the reconstruction error

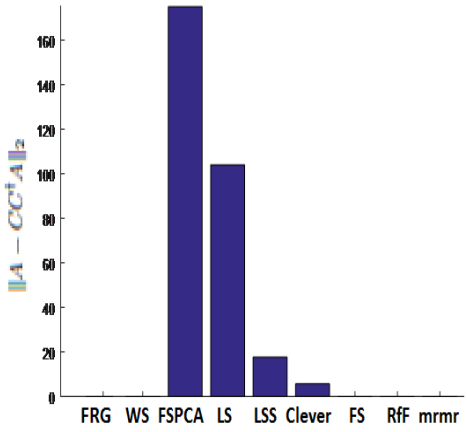

Figure 4.1: Madelon residuals minimization for 20 features

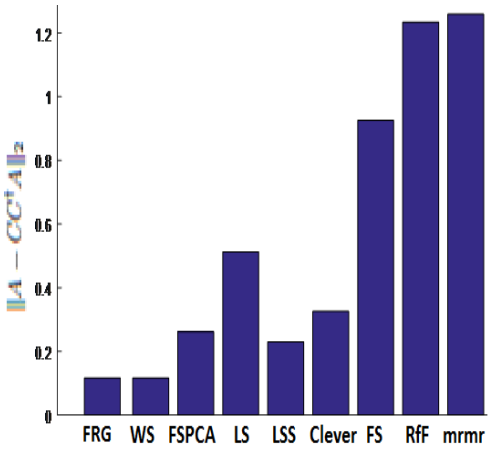

Figure 4.2: Inosphere residual minimisation for 5 features

Table 4.4: Madelon dataset 20 Most representative features selected using different algorithms

Techniques	20 Most Relevant Features Selected from Madelon																			
FRG $_{\epsilon>0.98}$	$P_{\mathbf{106},129}$		$P_{\mathbf{337},65}$		456	$P_{\mathbf{494},454}$		339	$P_{\mathbf{154},434,282}$			$P_{\mathbf{476},242}$		$P_{\mathbf{319},29,452}$			$P_{\mathbf{379},49}$		$P_{\mathbf{443},473}$	
WS	106	337	456	65	494	339	454	154	476	434	319	242	282	379	129	49	443	29	473	452
FSPCA	*223*	*256*	*407*	*251*	*331*	*18*	*148*	*322*	*188*	*221*	*406*	*200*	*413*	*243*	*313*	*52*	*402*	*334*	*350*	*286*
LS	*69*	*388*	*41*	*138*	*224*	*4*	64	*74*	*1*	*86*	*141*	*170*	*171*	175	*264*	*448*	*452*	*15*	*17*	*37*
LSS	*268*	*401*	*218*	*179*	*106*	*445*	*23*	*162*	*74*	*301*	*463*	*204*	*486*	*209*	*479*	*227*	*40*	*361*	*99*	*369*
Clever	29	106	282	339	*54*	*77*	*100*	107	*139*	*196*	*237*	*246*	*275*	*285*	*359*	*406*	*442*	*467*	*467*	*482*
FS	476	242	337	65	129	106	49	379	339	443	473	454	494	*324*	*425*	*206*	*412*	*205*	*283*	*297*
RfF	379	49	476	242	319	339	443	29	473	452	494	154	282	434	454	106	337	129	65	456
mrmr	242	49	129	443	337	379	454	65	473	494	339	106	*297*	*324*	*11*	*425*	*476*	*299*	*283*	*412*

$\|A - CC^{\dagger}A\|_{\xi}$ for a number of selected features between 5 and 35 from Madelon. We note a sharp decrease of the residual for selections between 5 and 20 features, and a more controlled decrease after 20 features, for Fisher Score, mrmr, ReliefF, WS and FRG. This behavior illustrates the progressive selection of very relevant features with much weight within the dataset. These observations are consistent with what is known of the dataset (it was generated with 20 relevant and 480 noise features). While the five techniques(mrmr, Fisher Score, ReliefF, WS and FRG) select some of the same features for feature subset sizes between 5 and 35, as seen on Fig. 4.3 , FRG and WS appear to be selecting the best sets each time, followed by ReliefF and mrmr.

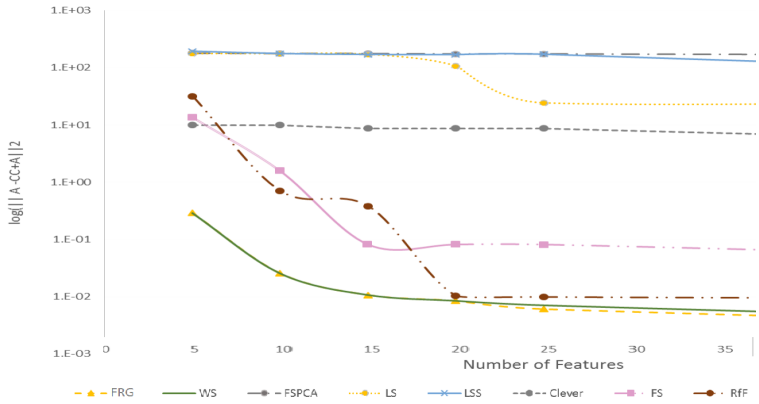

Figure 4.3: Reconstruction error $\|A - CC^{\dagger}A\|_{\xi}$ for selected features (5 to 35) on the Madelon dataset.

The Fisher Iris dataset is known to present a structure such that two of its features (the 3^{rd} and the 4^{th}) are sufficient to distinguish the underlying classes [66, 112]. As can be seen in Table 4.2, six of the compared techniques (including ours) accurately picked the 3^{rd} and the 4^{th} features as most relevant features.

The CorAl dataset characteristics are such as, 4 features are known to be relevant {1, 2, 3, 4}, one irrelevant feature {5} and one highly correlated {6} with a 25% error rate (matches the class label 75% of the time) [66]. Table 4.2 shows that of the compared methods, all but FSPCA [90] and the Leverage Score technique [42] consistently rank the 4 relevant features {1, 2, 3, 4} above 6 and 5.

We should note that, of the five feature selection techniques which performed well on these datasets, mrmr, Fisher, and ReliefF are supervised, while our techniques (WS and FRG) are unsupervised.

4.4.3.2 Classification Improvement with Feature Selection and Grouping

Our feature selection technique was also evaluated for classification performance on a large number of dataset.

Our feature selection technique was also evaluated for classification performance on a large number of dataset. In this specific experiment we looked to assess how

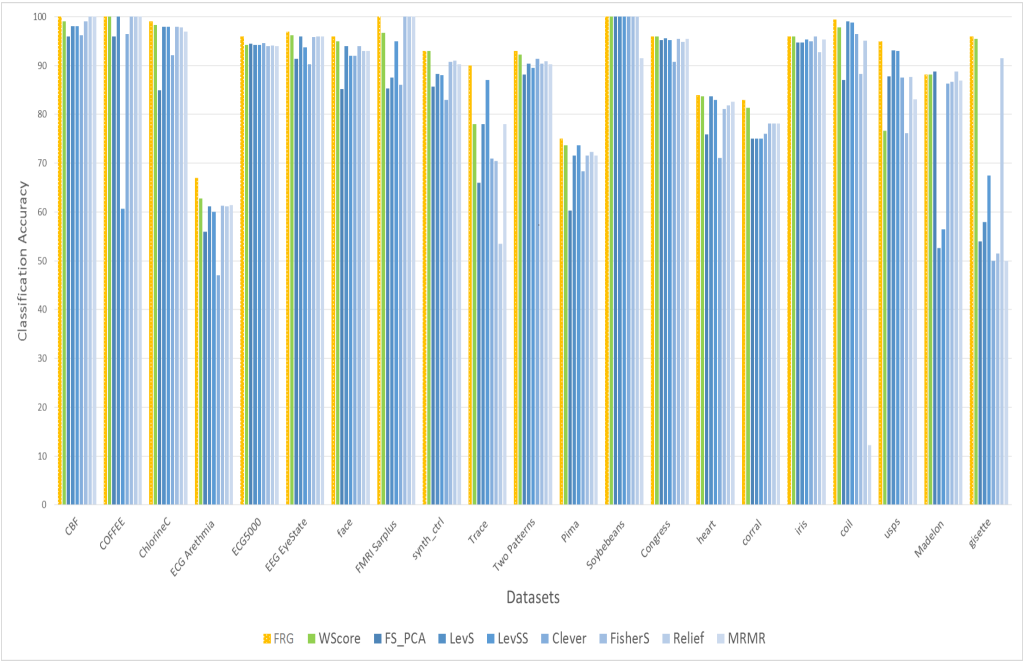

Figure 4.4: Classification accuracy of the nine techniques on a number of dataset.

the proposed feature selection technique contributes to improving classification. We used two classifiers from the Matlab Statistics and Machine Learning Toolbox: the linear Super Vector Machine (SVM) and Subspace K-Nearest-Neighbors (KNN). In most cases, as illustrated on Fig. 4.4, the classification accuracy for FRG improves when compared to that of WS. In applying correlation based feature grouping and reduction along with feature selection, FRG affords a framework with much more reduced computational cost and memory requirements for downstream applications while improving accuracy.

4.5 Summary

In this chapter we proposed an efficient unsupervised feature selection and grouping technique using PCA. It leverages the desired properties of the principal components by identifying the features that allow to retain the maximum variability of the data,

and hence results in reduced reconstruction error. In addition, our technique groups and re-combines correlated features to allow for additional insights in the data and enhanced accuracy for downstream pattern recognition tasks. It furthermore substantially reduces the dimensionality by removing a large number of misleading redundancies while retaining the important information for the learning process. Our experiments on numerous real datasets indicated the effectiveness of our technique. We are currently working on extending the proposed technique to application frameworks in which uncovering and grouping non-linearly dependent features is of interest.

Chapter 5: Transformation and Similarity Search

Multivariate time series (MTS) data mining has attracted much interest in recent years as increasing number of applications require the capability to manage and process large collections of MTS. In those applications, carrying out pattern recognition tasks such as similarity search, clustering, or classification can be challenging due to the high dimensionality, noise, redundancy, and feature correlated characteristics of the data. Dimensionality reduction is consequently often used as a preprocessing step to render the data more manageable. We propose in this chapter a novel MTS similarity search technique that addresses the challenge through dimensionality reduction and correlation analysis. An important contribution of the proposed technique is M2U, a technique allowing to transform an input MTS data with large number of variables to a univariate signal prior to searching for correlations within the set. The technique relies on unsupervised learning through PCA to uncover and use the weights associated with the original input variables in the univariate derivation. We conduct numerous experiments using various benchmark datasets to study the performance of the proposed technique. Compared to major existing techniques, our results indicate increased efficiency while providing improved similarity search accuracy. In what follows, we review the background and preliminaries in Section 5.1, discuss the related work in Section 5.2, and introduce the technique in Section 5.3. Section 5.4 presents the performance evaluation. A summary, our concluding remarks and future directions are presented in Section 5.5.

5.1 Background and Preliminaries

Innovation and advances in technology have led to the growth of data at a phenomenal rate. The existing MTS data reduction, analysis and mining techniques unfortunately do not scale well to its current challenges. The challenges include high dimensionality of the data, both in terms of the number of variables and the length of the time series, presence of noise and redundancies which make it difficult to uncover important patterns for many practical applications. Most pattern recognition tasks rely on dimensionality reduction as a crucial preprocessing step, for reasons of efficiency and interpretability, for a better understanding of the underlying processes that generated the data, and for easier downstream pattern recognition tasks.

The choice of reduction techniques requires careful considerations to ensure their suitability for the data at hand and downstream tasks, hence the effectiveness of the overall proposed technique. In similarity search for instance, when a much reduced representation is needed, MTS reduction techniques often follow one of three approaches. In the first approach, each variable within the MTS is considered independently as a time series [22], and often analyzed separately by using univariate techniques. While being easier to process, this approach often requires much more computation time. The second approach consists of stacking all data contained within all variables and form a UTS [46], to be analyzed as such using univariate techniques. Like the first one, this approach often overlooks the relationships that exist among the variables and cannot efficiently process a relatively large number of variables. The third approach, considers the MTS as a whole and transforms it into a lower dimensional representation that still captures its main characteristics, and the hidden relationships between the variables, while rendering the data more manageable. Although this approach presents more complexity, it provides more accurate results for the similarity search [86, 103, 82].

We propose a similarity search technique based on dimensionality reduction and time series correlations analysis. An important aspect of this technique is its representation basis on PCA which allows transforming the input MTS with large number

of variables to a UTS prior to looking for correlations. This is particularly important because, on one hand, the representation takes into account the correlations between variables within each multivariate dataset, in addition to decreasing redundancy, noise and, reducing its intrinsic high dimensionality. Other proposed univariate representations are often not able to retain the correlation between variables [22]. On the other hand substantial research and progress in making UTS pattern recognition tasks in general, and similarity search in particular, very efficient on large datasets has occurred in recent years [79, 12, 71]. Our proposed representation technique allows efficient UTS techniques to be easily extended and deployed to MTS.

5.1.1 Problem Formulation

A univariate time series (UTS) X = $< x_1, x_2, ..., x_n >$ of dimension n is a sequence of real values for a variable measured at n different timestamps. A MTS $A_{n,m}$ of n instances for m variables can be represented as an $n \times m$ matrix A (shown below) in which $a_{i,j}$ is the value of variable $X_{*,j}$ measured at time-stamp i, for $1 \leq i \leq n$, $1 \leq j \leq m$.

$$A_{n,m} = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,m} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,m} \end{bmatrix}$$

We are interested in the problem of similarity search in MTS defined as follows:

Definition 5.1. *(MTS similarity search)*
Let $D = \{A^1_{n,m}, A^2_{n,m}..., A^q_{n,m}\}$ be a set of MTS, each of which containing n instances and m variables; and ϵ be a user specified correlation threshold value. A MTS similarity search retrieves all pairs of times series A^i and A^j in D such that their coefficient of correlation is greater than ϵ, for $1 \leq i, j \leq q$.

Similarity search techniques in time series can be classified into two categories: subsequence search and whole sequence search. Here, we focus is on whole sequence

search and use Pearsons product-moment coefficient [76] as the measure for similarity between two time series.

5.1.2 Number of Principal Component to Retain

Algorithm 5.1 - Find the number k_{max} of PCs to retain

Input: $D' = \{A^1_{n,m}, A^2_{n,m}..., A^q_{n,m}\}$ a set of normalized MTS, θ cumulative variance to retain from each MTS.
Output: The number k_{max} of principal components to retain from each MTS s.t. $k_{max} = max(k_1, k_2, ..., k_q)$
begin

1: $k_{max} \leftarrow 0$
2: **for** $i \leftarrow 1$ *to* q **do**
3: Uncover fraction of total explained variance
4: f(k) $\leftarrow \Sigma_{z=1}^{k}\lambda_z / \Sigma_{z=1}^{r}\lambda_z$ for all z = {1, . . . , r}
5: Choose the smallest k so that f(k) $\geqslant \theta$ and retain that number of k eigenvectors to keep explained variance θ in the new embedding.
6: if $k > k_{max}$ then $k_{max} \leftarrow k$
7: **end for**
8: return k_{max}

end

In our proposed technique, we use a randomized version of the SVD technique [30]. To identify the number of principal components to retain from each MTS, we use the relative percentage variance criterion [41] to translate the amount of variance we wish to retain in the data to the number of principal components. The number k of relevant principal components may vary for different MTS, consequently k_{max}, representing the largest of all identified ks, are to be retained. Algorithm 5.1 summarizes the steps in uncovering k_{max}.

5.2 Related Work

Transforming MTS into lower dimensional time series has been an interesting research topic for which many dimensionality reduction methods have been proposed. Those

broadly adopted include Independent Component Analysis (ICA) [38], Random Projection (RP) [5, 24], and Principal Component Analysis(PCA) [94, 5, 24].

The Independent Component Analysis technique allows to find a new basis in which to represent the multivariate data. It can be considered a generalization of the PCA technique since the latter can be used as a preprocessing step in some ICA algorithms. However, while the goal in PCA in is to capture the maximum variance of data or minimize reconstruction error, the goal of ICA is to minimize the statistical dependence between the basis vectors. ICA however presents limitations that include the inability to determine the order of the independent components and the need for input time series data with non-Gaussian distribution.

The Random Projections technique is based on the Johnson Lindenstrauss lemma proposed in 1984 [40]. Unlike the PCA, it is not based on orthogonal transformations but rather on random projections to a lower dimension based on some distribution.

The lemma conveys that given a set of points in a high-dimensional space, they can be projected and embedded into a chosen much lower dimension subspace in such a way that distances between the points are nearly preserved. In the case of the random projection, the lower dimension where we look to embed the data is randomly chosen based on some distribution and we seek to have a probabilistic guaranty that the distance between two time series in the higher dimensional space will have some sort of correspondence with the distance between the same two series in the lower dimensional space. Considering a matrix $A_{n\times M}$ the original data with m variables and n observations, then $A_{k\times M} = R_{k\times n}A_{n\times M}$ is the random projection of $A_{n\times M}$ onto a lower k-dimensional subspace. This technique is carried out by projecting the original n-dimensional data onto a k-dimensional $(k << n)$ subspace, by using a random matrix whose rows have unit lengths $R_{k\times n}$. This data reduction technique is efficient for frameworks with a relatively small collection of very long time series length due to the fact that the data size k resulting from the reduction does not depend on the length of the time series but rather the number of time series [114]. It is however known to be less effective than PCA for severe dimensionality reduction [24].

The PCA technique is an orthogonal linear transformations in which one assumes all basis vectors to form an orthonormal matrix. It projects the original dataset in

a new coordinate system where the directions are pairwise orthonormal. A main advantage of PCA in our work is that it guaranties the uncovering of an optimal new embedding with minimal approximation error, and hence retains the crucial underlying structure of the original data. In addition to reducing dimensionality, the transformation decreases redundancy and noise, highlights relationships between the variables and reveals patterns by compressing the data while expressing it in such a way that highlights their similarity and dissimilarity. In addition, if two MTS are similar, their PCA representations will also be similar [51, 103]. Many similarity search techniques [103, 94, 4, 51] have relied on PCA for MTS processing as it is known to be one of the most efficiently computable techniques and a powerful tool of choice in high dimensional data environments for linear dimensionality reduction. PCA is however also limited by the fact that, as a new set of features is generated, the reduced form of the data is still a matrix. Retaining the first principal component in order to transform the data to a univariate signal has been explored with some level of success in the literature [94]. However, since principal components carry in decreasing order portions of the explained variance from the data, in order to retain enough information in the new representation, one would need to retain at least a few principal components in most cases. Hence the reduced form of the data would remain in a matrix form.

5.3 M2U Transformation

Given a set of MTS D= $\{A^1_{n,m}, A^2_{n,m}..., A^q_{n,m}\}$ and a user specified correlation threshold ϵ, our goal is to identify all pairs of time series in D whose Pearson correlation value is not less than ϵ. The proposed technique follows a two-steps resolution process. It first uses a novel transformation technique (M2U) to transform MTS to a UTS, then seeks pairwise correlations within the set of newly generated univariate series, using the Pearson product moment correlation. An important aspect about the proposed representation resulting from the M2U transformation is that it allows efficient UTS pattern recognition techniques to be easily extended to MTS.

5.3.1 M2U : Multivariate Time Series to a Univariate Time Series Transformation

In this section, we present the transformation process and describe its underlying intuition. Line 2 to 13 in Algorithm 5.2 provides the transformation steps.

Definition 5.2. *(Multivariate to Univariate Transformation (M2U)). Let* $A \in R^{n \times m}$ *be a matrix with rank* $r = rank(A)$ *s.t.* $r \leqslant min\{n, m\}$ *and* $k \leqslant r$*. Let* V_k *be the matrix containing the top k right singular vectors of A and* S_k *be the matrix containing the top k singular values of A. Then, the (rank-k) univariate representation of A is defined as* $[U_{n,1}]_i^k = \Sigma_{v=1}^{m} a_{i,v} \hat{w}_v$ *, for* $i = 1, 2, ..., m$*, where:*

- $a_{i,v}$ *is the element of matrix A at row i, and column v.*
- $\hat{w}_j = \Sigma_{z=1}^{k} w_z e_{j,z}$*, for* $j = \{1, 2, ..., m\}$ *is the weight of the column variable j within the given multivariate dataset, called weighted score defined below.*
- $[U_{n,1}]_i^k = \Sigma_{v=1}^{m} a_{i,v} \hat{w}_v$ *is the i-th entry of the newly generated UTS* $U_{n,1}$*.*

We assume that each MTS $A_{n,m}^i$ in D of n instances for m variables can be represented as an $n \times m$ normalized matrix A (shown below).

Each column variable $X_{*,j}$ holds a particular weight or importance $\hat{w}_j$ with respect to the whole data matrix $A_{n \times m}$ [48]. Let us consider $\hat{w}$, the weight vector containing all variable weights.

$$A_{n,m} = \begin{array}{c} \begin{matrix} X_{*,1} & X_{*,2} & \cdots & X_{*,m} \end{matrix} \\ \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,m} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,m} \end{bmatrix} \end{array}$$

Intuitively, to transform a MTS to a UTS in a new framework, we will need to uncover and take into account the variable's importance or weight in the reconstruction process.

Algorithm 5.2 - M2U and Pairwise Correlation Search

Input: $D' = \{A^1_{n,m}, A^2_{n,m}..., A^q_{n,m}\}$ a set of normalized MTS, θ (cumulative variance explained), ϵ a user specified Pearson correlation threshold.
Output:A set C of all pairs (A^i, A^j) in D' whose correlation is not less than ϵ.
begin

Estimate k_{max} using Algorithm 5.1.
  $k \leftarrow k_{max}$
for $i \leftarrow 1$ *to* q **do**
  STEP1: Reduce MTS $A^i \in D'$ **to UTS** U^i**, add it to** D^U
    $A \leftarrow$ *the* i^{th}MTS of rank r, in D', $A^i_{n,m}$
    Compute the Singular Value Decomposition
    $[U, S, V^T] \leftarrow SVD(A)$
    Retain a matrix of k eigenvectors
    $M \leftarrow V_k$
    Build the weighted matrix $[wV_k]$
    For z $\leftarrow$ 1 to k
      $w_z \leftarrow \lambda_z / \Sigma^r_{z=1} \lambda_z$
      $[wV_k]_{*,z} \leftarrow w_z * [M]_{*,z}$
    end for
    Compute the weighted score for each variable
      $\hat{w}^{(k)}_j \leftarrow |\Sigma^k_{z=1} w_z e_{j,z}|$, for all j = {1, 2, ..., m}.
    Build the weighted matrix $[\hat{w}A]$
    For v $\leftarrow$ 1 to m
      $\hat{w}_v \leftarrow \hat{w}^{(k)}_v$
      $[\hat{w}A]_{*,v} \leftarrow [A]_{*,v} * \hat{w}_v$
    end for
    Uncover row entries for the new univariate signal $U_{n,1}$
      $[U_{n,1}]_i \leftarrow \Sigma^m_{v=1} a_{i,v} \hat{w}_v$, for i = {1, 2, ..., n}
    add $[U_{n,1}]$ to D^U
end for
STEP2: Uncover correlated pairs
  For all $(U^i, U^j) \in D^U$
    Compute their pairwise Pearson correlations
    If ($|\rho(U^i, U^j)| \geqslant \epsilon$) then add (A^i, A^j) to C

end

5.3.1.1 Finding the Weighted Scores(Variable Weights)

We rely on unsupervised learning through a principal component analysis of the input data to uncover the variable weights (weighted scores) within $\hat{w}$. We use information drawn from the diagonal of the matrix S and the rows of matrix V (from the factorization $A = USV^T$) to computed statistics that reveal influence on the columns of the original matrix A.

Let us first note that the entries in each column of $V = A^T US^\dagger$ (where $S^\dagger$ denotes the Moore pseudo-inverse of S) provide the regression coefficients of a corresponding principal component, which in turn is expressed as a linear combination of all variables from the original matrix. More precisely, the coefficient of the i^{th} new feature component uncovered through PCA is expected to be the i^{th} entry of the eigenvector. The first k principal components can be expressed as follows, where $X_1, ..., X_m$ are the original variables within the data matrix A.

$e_{1,1}X_1 + e_{1,2}X_2 + e_{1,m}X_m = PC_1$
$e_{2,1}X_1 + e_{2,2}X_2 + e_{2,m}X_m = PC_2$
...
$e_{k,1}X_1 + e_{k,2}X_2 + e_{k,m}X_m = PC_k$

Just as the principal components can be expressed as a linear combination of all the variables from the original matrix, the original variables can also be defined as linear combinations of the principal components. The rows of V each concern a specific variable and are considered rescaled data projected onto the principal components; the data is indeed rescaled according to the singular values to ensure that the covariance is identity.

In the multivariate to univariate transformation process, we wish to uncover the influence of the original variables with respect to the input data. Thus, we will seek to retain coefficients that are "unscaled". Such coefficients will account for the relative portions of variance carried by the principal components.

Definition 5.3. *(Weighted Scores) Let $A \in R^{n \times m}$ be a matrix with rank $r = rank(A)$ s.t. $r \leqslant min\{n, m\}$ and $k \leqslant r$. Let V_k be the matrix containing the top k right singular*

vectors of A and S_k be the matrix containing the top k singular values of A. Then, the (rank-k) weighted score of the i-th column of A is defined as $\hat{w}_i^{(k)} = |\Sigma_{j=1}^{k} w_j e_{i,j}|$, for $i = 1, 2, ..., m$.
where:

- $w_j = \lambda_j / \Sigma_{z=1}^{r} \lambda_z$, *the fraction of variance carried by the j-th column in* $[V_k]$, *for* $1 \leqslant j \leqslant k$ *and*
- $\lambda_j = \sigma_j^2/(n-1)$ *is the variance corresponding to the* j^{th} *singular value*(σ_j), *consequently to the* j^{th} *column of* $[V_k]$, *and* $\lambda_1 \geq \ldots \geq \lambda_r \geq 0$.

Let us note that the weight w within the weighted score is reflected through the proportion of explained variance retained by the specific principal component. For instance if we consider the j^{th} principal direction, its weight labeled w_j is $w_j = \lambda_j / \Sigma_{z=1}^{r} \lambda_z$. A matrix $[wV_k]$ of weighted principal directions is then constructed by multiplying each component within the retained matrix of eigenvectors V_k by its corresponding weight w_j.

$$\begin{matrix} & E_1 & E_2 & \cdots & E_k \end{matrix}$$

$$V_k = \begin{bmatrix} e_{1,1} & e_{1,2} & \cdots & e_{1,k} \\ e_{2,1} & e_{2,2} & \cdots & e_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ e_{m,1} & e_{m,2} & \cdots & e_{m,k} \end{bmatrix}$$

$$\begin{matrix} & w_1E_1 & w_2E_2 & \cdots & w_kE_k \end{matrix}$$

$$[wV_k] = \begin{bmatrix} w_1e_{1,1} & w_2e_{1,2} & \cdots & w_ke_{1,k} \\ w_1e_{2,1} & w_2e_{2,2} & \cdots & w_ke_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ w_1e_{m,1} & w_2e_{m,2} & \cdots & w_ke_{m,k} \end{bmatrix}$$

Subsequently, the row entries of the weighted matrix $[wV_k]$ are aggregated as per line 9 of Algorithm 5.2 to provide the variable weights vector $\hat{w}$.

$$\hat{w} = \begin{bmatrix} |w_1e_{1,1} + w_2e_{1,2} + \cdots + w_ke_{1,k}| \\ |w_1e_{2,1} + w_2e_{2,2} + \cdots + w_ke_{2,k}| \\ \vdots \quad \vdots \quad \ddots \quad \vdots \\ |w_1e_{m,1} + w_2e_{m,2} + \cdots + w_ke_{m,k}| \end{bmatrix} = \begin{bmatrix} \hat{w_1} \\ \hat{w_2} \\ \vdots \\ \hat{w_m} \end{bmatrix}$$

The variable weights vector $\hat{w}$ entries expressed as $\hat{w_j} = |\Sigma_{z=1}^{k} w_z e_{j,z}|$, for $j = \{1, 2, ..., m\}$,

are the original weights for the column-variables within the given multivariate dataset.

5.3.1.2 Deriving the Univariate Signal

Once the variable weights are uncovered, the next step consists of building a weighted matrix $[A\hat{w}]$ by factoring the original data matrix $A_{n\times m}$ and the variable weights vector $\hat{w}$. More precisely, as shown on lines 10 and 11 of Algorithm 5.2, each column of $A_{n\times m}$ is factored by its corresponding weight and the row entries of the weighted matrix are subsequently aggregated to form the new univariate derivation.

$$U_{n,1} = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,m} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,m} \end{bmatrix} * \begin{bmatrix} \hat{w_1} \\ \hat{w_2} \\ \vdots \\ \hat{w_m} \end{bmatrix} = \begin{bmatrix} \sum_{v=1}^{m} a_{1v}\hat{w_v} \\ \sum_{v=1}^{m} a_{2v}\hat{w_v} \\ \vdots \\ \sum_{v=1}^{m} a_{nv}\hat{w_v} \end{bmatrix}$$

An important aspect of this representation technique is that it uses statistics drawn from the PCA to leverage the relative importance of each variable and uncovers a univariate derivation of the time series. The new derivation takes into account the correlation between variables in the MTS dataset and, decreases redundancy and noise. The proposed representation will allow efficient UTS pattern recognition techniques to be easily extended to MTS.

5.3.2 Similarity Measure

We use Pearson's product-moment coefficient [76] as the measure to assess similarity between two time series. The Pearson correlation measure is known to be more robust against data that is not normalized and to respond better to baseline and scale shifts when compared to other measures [114].

Let X and Y be two normally distributed time series of equal dimension n. The Pearson correlation coefficient of X and Y denoted $\rho(X, Y)$, is a value in [-1,1] that

measures the linear dependency between X and Y, defined as follows:

$$\rho(X,Y) = \frac{\sum_{t=1}^{n}(x_t - \overline{x})(y_t - \overline{y})}{\sqrt{\sum_{t=1}^{n}(x_t - \overline{x})^2}\sqrt{\sum_{t=1}^{n}(y_t - \overline{y})^2}} \tag{5.1}$$

where $\overline{x}_t$ is the mean of X over n and $\overline{y}$ is the mean of Y over n. The Pearson correlation coefficient can be approximated to the Pearson product moment, expressed as follows:

$$\rho(X,Y) = \frac{1}{n-1}\sum_{t=1}^{n}\frac{xy}{S_x S_y} \tag{5.2}$$

where $x = (x_t - \overline{x}), y = (y_t - \overline{y})$, $S_x = [(1/n-1)\sum_{t=1}^{n} x^2]^{1/2}$, and $S_y = [(1/n-1)\sum_{t=1}^{n} y^2]^{1/2}$.

Given a user specified correlation threshold ϵ, our goal is to identify all pairs of time series whose Pearson correlation value is not less than ϵ. Algorithm 5.2 summarizes the steps for the pairwise correlation search from line 14 to 17.

5.4 Performance Evaluation

Our proposed technique was implemented in Matlab and, numerous experiments we conducted on benchmark datasets, using a PC configured with Intel Quad core i7 2.00 GHz CPU, 8 GB RAM and, running Windows 7.

5.4.1 Benchmark Datasets

The experiments were ran on benchmark datasets drawn from widely used repositories [3, 55, 39]. We present experiments and results for three of these benchmark datasets used.

The Australian language sign dataset(AUSLAN) [45] was gathered through two gloves, with 22 sensors while native AUSLAN speakers signed. The dataset contains 95 signs having 27 examples each, hence a total of 2565 of signs gathered. This dataset is well used in similarity search problems due to its complexity.

The INRIA Holidays images dataset (INRIA HID) [39] is a collection of images used in testing robustness to various transformations: rotations, viewpoint and illumination changes, blurring, etc. The dataset contains 500 high resolution image groups representing a large variety of scene types to incorporate diversity in representation.

The Transient classification benchmark dataset (Trace) [83] was gathered for power plant diagnostics. The dataset has 5 variables (4 process variables and a class label) and 16 operating states. The class label is set to 0 until the transient occurs, at which time it is set to 1. The part of the data that is of interest in our work is the subset where the transient occurs.

5.4.2 Evaluation and Results

We designed experiments to assess the performance of the proposed technique. In this section, we compare our performance against those from primarily five other techniques: the Correlation Based Dynamical Time Warping (CBDTW) [4], the 2-D correlation measure for matrices(see section 6.2) (*Corr2*), the Dynamical Time Warping(DTW), Eros [106] and the Euclidien Distance(ED).

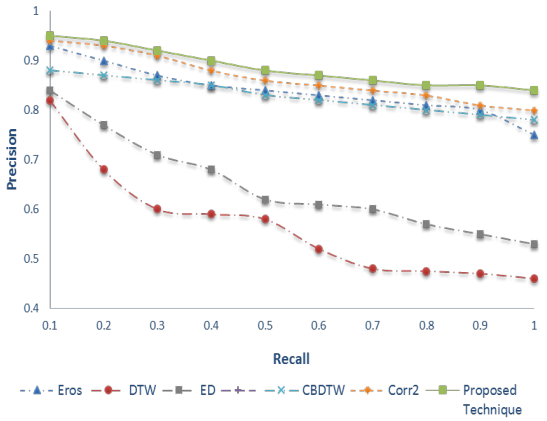

Figure 5.1: Recall-Precision on AUSLAN

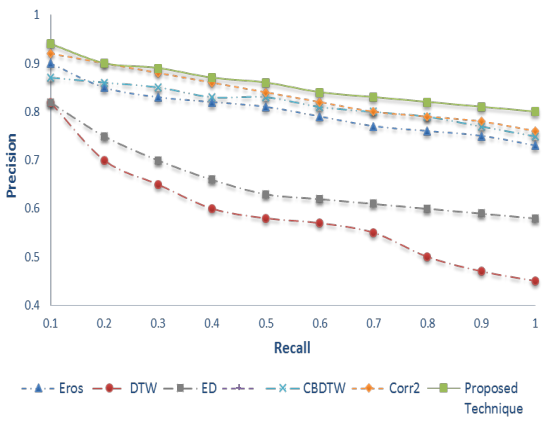

Figure 5.2: Recall-Precision on TRACE

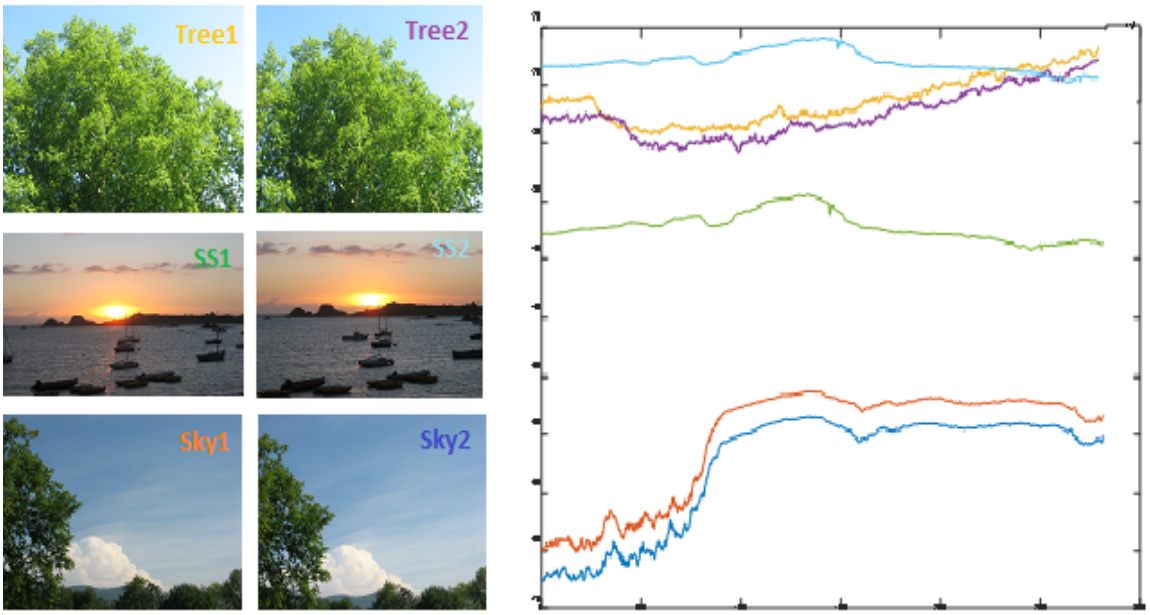

Figure 5.3: Left -Six images from the INRIA HID of three scenes taken at different points in time, found as closest matches. Right -Univariate signals for the six images after M2U transformation. Image names are color-coded with their corresponding signal.

The recall-precision ratios recorded for all techniques on the datasets AUSLAN and TRACE datasets are shown in Fig. 5.1 and Fig. 5.2 respectively. On both datasets, we can see that the Euclidien Distance(ED) and Dynamical Time Warping(DTW) perform worst compared to other techniques. This may be due to the fact that, neither of these techniques takes into account the existing correlations between the variables of the MTS while the remaining four techniques do. Our technique outperforms the remaining techniques on both datasets. In another set of experiments, we further evaluated how the proposed univariate representations compares to the case where the original matrices are used to find pairwise correlations within a set of MTS. Our results confirm that our technique yields improved similarity search accuracy. To illustrate this, let us consider the six images from the INRIA Holidays images dataset,of three scenes taken at different points in time, on the left side of Fig. 5.3.

For the purpose of the experiment, the images were converted to the grayscale intensity images, then to double precision to transform the true-color image RGB to 2-dimensional matrices. Each image is represented by a 2816×2112 matrix.

Using our proposed transformation technique M2U, each matrix is transformed into a univariate signal represented on the right side of Fig. 5.3. The color of each univariate signal (Fig. 5.3 right) matches the color of the text on its corresponding image to the left side (Fig. 5.3 left). We can see that similar images generated similar univariate signals. Furthermore, using our technique, the Pearson correlation coefficients post transformation are:

ρ(Tree1,Tree2)=0.9661,
ρ(SS1,SS2)=0.9413,
ρ(Sky1,Sky2)=0.9982.

To uncover the correlation coefficient obtained using the original matrices without transformation, we use the 2-D correlation coefficient *Corr2*, defined as:

$$Corr2(A^i, A^j) = \frac{\Sigma_n \Sigma_m (A^i_{mn} - \bar{A}^i)(A^j_{mn} - \bar{A}^j)}{\sqrt{(\Sigma_n \Sigma_m (A^i_{mn} - \bar{A}^i)^2)(A^j_{mn} - \bar{A}^j)^2}}$$

where $\bar{A}^i = mean2(A^i)$ and $\bar{A}^j = mean2(A^j)$.

For this set of experiments on the full image matrices, the Pearson correlation coefficients are:

Corr2(Tree1,Tree2)=0.6261,
Corr2(SS1,SS2)=0.7594,
Corr2(Sky1,Sky2)=0.8027.

Let us consider the case in which we are looking for similar images with a correlation coefficient greater than a correlation threshold $\epsilon = 0.7$. In this case, the images of Tree1 and Tree2 would not have been returned as correlated if the full matrix is used, while it would be identified if the Pearson correlation is applied to the univariate

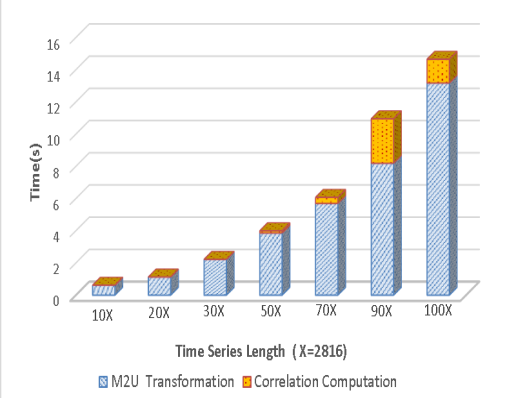

Figure 5.4: Runtime for each step in the proposed technique as the length of the time series increases (INRIA dataset)

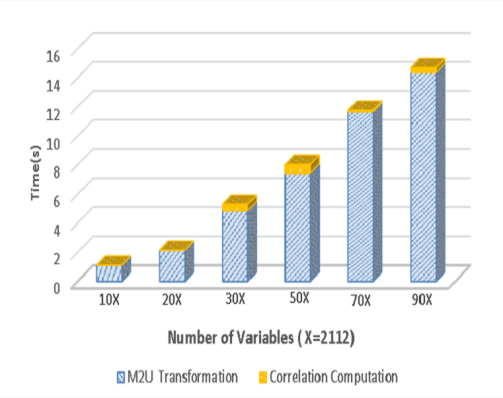

Figure 5.5: Runtime for each step in the proposed technique as the number of variables increases (INRIA dataset)

derivation using our technique (M2U).

Transforming MTS to a univariate signal yields improved similarity search accuracy. When the matrix goes through a PCA transformation, in addition to reducing the dimensionality, it decreases redundancy and noise, highlights relationships between the different variables, and reveals patterns by compressing the data while expressing it in such a way that highlights their similarity and dissimilarity. In addition, since we are not discarding any of the relevant principal components, but rather re-combing variables, we preserve much of the relevant and needed information from the data.

In this set of experiments, we concatenate images from the INRIA dataset to build larger images, hence larger matrices to compare for scalability. As the length of the time series or the number of variables grows, the runtime grows as well. We can identify two phases within the runtime as illustrated in Figures 5.4 and 5.5 for the INRIA dataset. The first phase (M2U transformation) allows using the M2U algorithm and represent the MTS as a UTS. This first step ultimately allows for a tremendous saving in memory space, and in computation time for downstream applications. The second phase allows computing the correlation values and identify similar time series. Of the two phases, we note that the transformation step makes

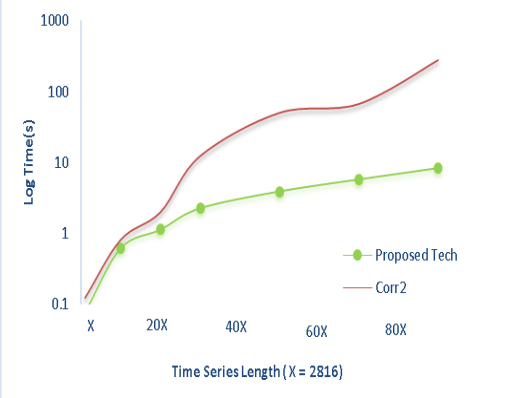

Figure 5.6: Comparing the proposed technique runtime to that of *Corr2* as the length of the time series varies

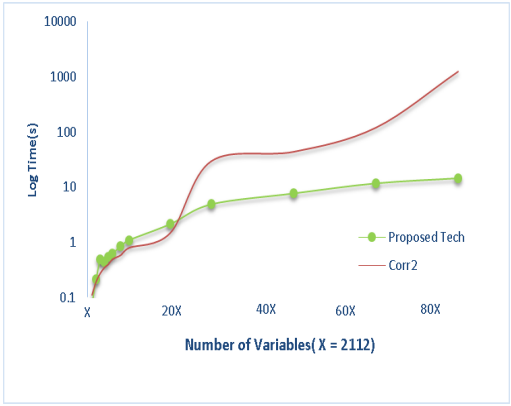

Figure 5.7: Comparing the proposed technique runtime to that of *Corr2* as the number of variables varies

up a larger portion of the run time. This is due to the fact that, once the multivariate series is reduced to a univariate series, the second step will merely consist of finding the correlation value between two UTS. We remark that the runtime in the first step has also been substantially reduced using the Randomized SVD technique rather than using the standard SVD. To illustrate the difference in computation time between the two versions of SVD, let us consider a matrix of size 650×497 to be reduced using standard SVD and Randomized SVD. The reduction time using standard SVD is about 0.11061 seconds with an SVD error of 0.1913, while the reduction time using randomized SVD is about 0.0065 seconds with SVD error of 0.1921.

The proposed technique performs sizeably better than peer techniques in terms of computation time, Fig. 5.6 and Fig. 5.7 illustrate respectively the comparison of the runtime between our proposed technique to that of *Corr2* as the length (Fig. 5.6) and number of variables (Fig. 5.7) of the time series vary. In both cases we use logarithmic time scale and illustrate the time increase as the dimensionality(dimensions and variables) increased. We can note from Fig. 5.6 and Fig. 5.7 that the proposed technique only slightly outperforms *Corr2* in terms of runtime for low number of variables and instances (lower than 20X, where X is the number of variables or instances). However, for a number of variables over 20X, the proposed technique significantly outperforms

Corr2. For instance, at 60X, the computation time of *Corr2* is an order of magnitude greater than that of the proposed technique and on average 85 times the runtime of our proposed technique for 90X.

5.5 Summary

We proposed a novel technique for MTS transformation, analysis and search. The technique relies on dimensionality reduction and correlation analysis to uncover similar MTS. It uses statistics drawn from the Principal Component Analysis to find a unique conversion of MTS to UTS representation prior to seeking correlations. Our experiment results indicate increased accuracy and efficiency when compared to major existing techniques. The proposed representation allows efficient techniques for UTS to be easily extended to MTS.

Chapter 6: Trend and Value based Representation and Similarity Search

Research in time series knowledge discovery in general, and in similarity search in particular has been very active in recent years, due to the number of application domains that are progressively requiring to work with large amounts of high dimensional time series data. Unfortunately, for many practical applications, high dimensionality of data in such frameworks makes it difficult to uncover important patterns from the raw data. Hence time series transformation techniques have become important preprocessing tools for many pattern recognition tasks. In this chapter we investigate the problem of similarity search in time series and propose a symbolic transformation technique that incorporates the time series value and trend information to enhance accuracy in the search results; and a symbolic similarity measure. We also apply this technique along with the representation technique M2U from section 5.2 to multivariate time series(MTS) datasets to investigate the problem of similarity search for MTS. We conduct numerous experiments to evaluate the performance of the proposed technique. Our results indicate increased accuracy and efficiency compared to existing techniques. In what follows, we review the background and preliminaries in section 6.1, discuss the related work in Section 6.2, and introduce the technique in Section 6.3. Section 6.4 introduces an application of the proposed technique to MTS. Section 6.5 presents the performance evaluation. A summary, and future directions

are presented in Section 6.6.

6.1 Background and Preliminaries

Innovation and advances in data generation and collection technologies have led to the growth of data at a phenomenal rate. This presents continuous challenges to existing data exploration, analysis and mining techniques. This is true in particular for time series, which increasingly makes up a large fraction of the world's supply of data. Many fields such as medical monitoring and imaging, aerospace science, and Finance, require the ability to manage and process large collections of time series data and to discover interesting and meaningful patterns in the data.

Research on time series repositories and streaming systems has been very active in recent years, looking to improve existing models and techniques and/or develop new ones that would gracefully scale to today's needs. This has resulted in much progress, particularly for univariate time series (UTS) where substantial research in making time series search very fast on very large datasets has occurred [79, 88, 12]. There is however a need to further improve the results to better address todays evolving data needs in terms of volume, velocity, veracity, value, and for efficient analysis and search. In this chapter we propose a similarity search technique based on time series correlations analysis. The technique looks through a dataset of time series with large number of instances and returns pairs of correlated time series without having to go through a pairwise comparison of all the series in the dataset.

Our contributions can be summarized as follows:

- A symbolic representation technique incorporating trend and value information from the original time series to better capture and represent its characteristics.
- The formulation of a similarity measure based on a binary weighted dissimilarity measures for mixed types of variables measuring different objects to improve accuracy.

6.2 Preliminaries

In this section we review some background, definitions and notions needed in this chapter.

A UTS X = $< x_1, x_2, ..., x_n >$ of dimension n is a sequence of real values for a variable/attribute measured at n different timestamps. We are interested in the problem of similarity search in time series defined as follows:
Let $D^U = \{X^1, X^2 ..., X^q\}$ be a set of UTS of n instances each, and ϵ be a user specified correlation threshold value. A time series similarity search retrieves all pairs of times series X^i and X^j in D^U such that their correlation distance does not exceed ϵ, for $1 \leq i, j \leq q$.

Similarity search techniques in time series can be classified in two categories: subsequence search and whole sequence search. In our work,we focus on whole sequence search for which we use Pearson's product-moment coefficient [76] as the measure to assess similarity between two time series. This measure is known to be more robust against data that is not normalized and to respond better to baseline and scale shifts when compared to other measures [114].

Let X and Y be two normally distributed time series of equal dimensions n. The Pearson correlation coefficient of X and Y denoted $\rho(X, Y)$, is a value in [-1,1] that measures the linear dependency between X and Y, defined as follows:

$$\rho(X,Y) = \frac{\sum_{t=1}^{n} (x_t - \overline{x})(y_t - \overline{y})}{\sqrt{\sum_{t=1}^{n} (x_t - \overline{x})^2} \sqrt{\sum_{t=1}^{n} (y_t - \overline{y})^2}} \tag{6.1}$$

where $\overline{x}_t$ is the mean of X over n and $\overline{y}$ is the mean of Y over n.

The Pearson correlation coefficient can be approximated to the Pearson product moment as follows:

$$\rho(X,Y) = \frac{1}{n-1} \sum_{t=1}^{n} \frac{xy}{S_x S_y} \tag{6.2}$$

where $x = (x_t - \overline{x}), y = (y_t - \overline{y})$, $S_x = [(1/(n-1)) \sum_{t=1}^{n} x^2]^{1/2}$ and,
$S_y = [(1/(n-1)) \sum_{t=1}^{n} y^2]^{1/2}$.

Table 6.1: Notations used in this chapter

D	Set of MTS (or, D' if if the time series are normalized)
D^U	Set of UTS, D'^U if normalized.
$\hat{D^S}$	Set of symbolic time series resulting from the STEP1 transformation
X	UTS data $X = < x_1, x_2, ...x_n >$
$\hat{X}$	A symbolic representation of the time series X
θ	The explained variance in the data that are represented within k retained principal components
ρ	Correlation coefficient
$\hat{\rho}$	Symbolic correlation coefficient
ϵ	User specified Pearson correlation threshold
C	Resulting correlation set

6.3 Related Work

Processing raw time series data in similarity search presents major challenges due to the high dimensionality, noise, redundancy and feature correlated characteristics of the data. One way to reduce those challenges is to rely on reduction and representation techniques that provide a synopsis structure while retaining important characteristics of the data. As the first step in our proposed technique, we discretize the UTS into a symbolic string to ease its processing. Numerous such representation techniques have been proposed in recent years [67, 70, 81, 111, 89]. While some of these techniques are based on the "clipping" or hard limiting technique [81, 53, 54], for which the compression ratio is data dictated, most of them are data adaptive and assume that the user has a choice over the compression ratio of the data when generating the new representation. Among them, the Symbolic Aggregation Approximation (SAX) transformation [67] has particularly been widely used in the literature due to its simplicity and ability to transform the time series data without much knowledge about the data. Through a two steps process: a reduction based on the

Piecewise Aggregation Approximation(PAA) [107, 57] and a discretization, it allows a time series of some length n to be reduced to a string of arbitrary length ω, ($\omega <$ n), using an alphabet of arbitrary integer size a, where typically $a > 2$. The original time series is discretized by first obtaining a PAA approximation and subsequently using predetermined breakpoints to map the PAA coefficients to SAX symbols.

The clipping technique provides an actual bit level approximation of the data where each bit indicates whether the series is above or below the average. It is a particular case of the SAX transformation technique with two classes and no dimension reduction [81].

An important advantage of the clipping technique over other techniques is that it allows comparing the original raw data directly to the new representation and satisfies the lower bounding property [81]. However, like the SAX transformation technique, clipping is limited as it does not take into consideration the trend information from the original time series. Recent research [111, 69] in time series representation have incorporated both the value and trend to improve the approximation accuracy. While progress has been made, improvements are still needed.

In [69] the technique proposes 1D-Sax, an extension to the SAX transformation, which incorporates the trend information and represents the time series in binary sequence. The time series is first divided into segments on which the corresponding linear regression of the time series is computed and eventually quantized into a symbol combining both the trend and value. Hence, while it improves on the retention of the original time series information, it does not provide the flexibility to reflect the weight that the trend or the value should bear based on the original time series structure. Zhang et al. in [111] present a technique that takes into account the trend information and adapts to streaming time series. The technique transforms each real value into a binary bit for the value, and after a first pruning phase, also transform the direction on a wider interval into one bit. This technique goes through two full steps of transformations and candidate set selection prior to computing the similarity measure on the reduced set of original time series. In large data setting going through those steps can lead to much overhead in terms of computation time. Our proposed method transforms the time series into a binary representation that incorporates

both the value and trend information from the original series by using the Piecewise Aggregation Approximation (PAA) and extending the clipping technique [81].

6.4 Trend and Value Based Representation

Our proposed technique follows a two steps solution process. The first step, TVR(Trend and Value Representation) uses a novel symbolic time series representation technique which extends the Clipping technique [81]. The representation incorporates the time series trend information, in addition to the value information in order to better capture the time series characteristics while providing greater accuracy and flexibility. The advantage of this representation is that it provides dedicated bit positions for the value and trend, and, hence allows more flexibility to calibrate either one without affecting the other.

In the second step, we use our proposed weighted symbolic correlation measure for mixed types of binary variables measuring different objects, and a preset threshold to allow identifying a reduced candidate set before computing pairwise correlation using Pearson correlation. The symbolic correlation measure provides greater accuracy and flexibility for frameworks for which either the trend or the value based approach would be better suited. Lines 1 to 8 of Algorithm 6.1 represent step 1 (TVR), while step 2 is presented from lines 9 to 15. In what follows, we explain details of our the proposed solution.

Given a correlation threshold ε, our goal is to determine all pairs of correlated times series, with a correlation greater than ϵ, within the dataset, while avoiding the costly pairwise comparisons in large datasets. In this step, we first discretize the UTS into a symbolic string. The UTS is then transformed into a boolean symbolic representation of size 2ω, with ω being the number of PAA segments.

The information within each segment will be represented by two bits.
For i= 1 to 2ω;

- the 1^{st} bit represents the value information, hence the position of the current point of the series in relation with its mean

Algorithm 6.1 - Correlated Trend Value Representations(CTVR)

Input: $D'^U = \{X^1, X^2..., X^q\}$ a set of normalized UTS of n instances each, ϵ a user specified Pearson correlation threshold.

Output: A correlation set C of all pairs (X^i, X^j) in D' with correlation greater than ϵ.

begin

STEP1 - TVR : Time Series Transformation
for $i \leftarrow 1$ *to* q **do**
  Transform X^i to a symbolic series $\hat{X}^i$ of size 2ω and add it to $\hat{D}^S$
    Compute the mean value of each the Series
    $\bar{X}^i \leftarrow 1/n \sum_{z=1}^{n} x_{zi}$
    For z $\leftarrow$ 1 to ω
      If $\bar{x}_{z,i} \geq \bar{X}^i$ then $\hat{X}_{2z-1} \leftarrow 1$ else $\hat{X}_{2z-1} \leftarrow 0$
      if $\bar{x}_{z,i} \leq \bar{x}_{z+1,i}$ then $\hat{X}_{2z} \leftarrow 1$ else $\hat{X}_{2z} \leftarrow 0$
    end for
end for
STEP2: Uncover correlated pairs
  For all $\hat{X}^i, \hat{X}^j \in \hat{D}^S$,
    Compute their pairwise symbolic correlations
    If $(|\hat{\rho}(\hat{X}^i, \hat{X}^j)| \geqslant \varepsilon)$ then add (X^i, X^j) to C_S
  For all $(X^i, X^j) \in C_S$
    Compute their pairwise Pearson correlations
    If ($|\rho(X^i, X^j)| \geqslant \epsilon$) then add (X^i, X^j) to C

end

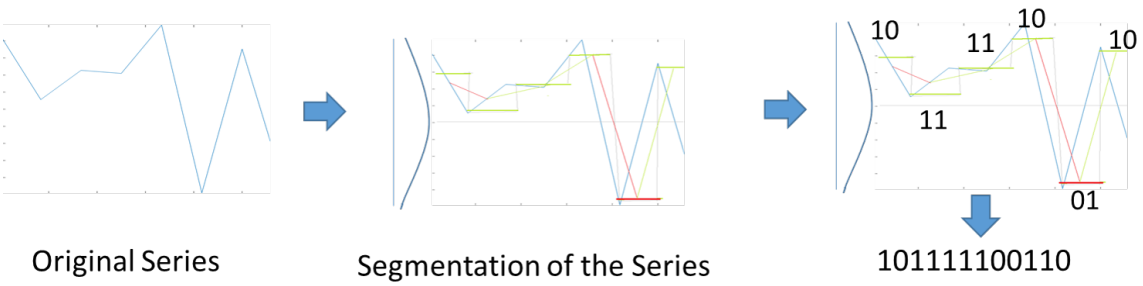

Figure 6.1: Steps in transforming UTS into symbolic strings using our proposed technique.

- the 2^{nd} bit represents the current trend of the series compared to its most recent position

More formally stated:
Let $X = <x_1, \ldots, x_n>$ be a time series of length n. The corresponding symbolic Boolean series is $\hat{X} = <\hat{s}_1, \ldots, \hat{s}_{2\omega}>$ of length 2ω where ω is the number of segments. The symbolic series $\hat{X}$ combines boolean values characterizing the position of the point compared to the mean of the series (in the odd positions within $\hat{X}$), and for the local trend, the position of the midpoint of the current segment is compared to the most recent midpoint ($\bar{x}$) encountered (in the even positions within $\hat{X}$).

For i= 1 to ω:

$$\hat{X}_{2i-1} = \begin{cases} 1\ , if\ \bar{x}_i \geq\ \bar{X} \\ 0, if\ \bar{x}_i <\ \bar{X} \end{cases} \qquad \bar{X} = \frac{1}{n} \sum_{t=1}^{n} x_t,$$

$$\hat{X}_{2i} = \begin{cases} 1, \quad if\ \bar{x}_i\ \leq \bar{x}_{i+1} \\ 0,\ if\ \bar{x}_i > \bar{x}_{i+1} \end{cases}$$

Fig. 6.2 illustrates the proposed representation technique in the particular case where ω (the number of segments) is equal to n (the length of the original time series). In this case, each numerical value of the series is transformed into a symbol illustrating its position in relation with the mean. A second symbol reflecting the direction is allocated as a result of comparing two consecutive values to get a sense

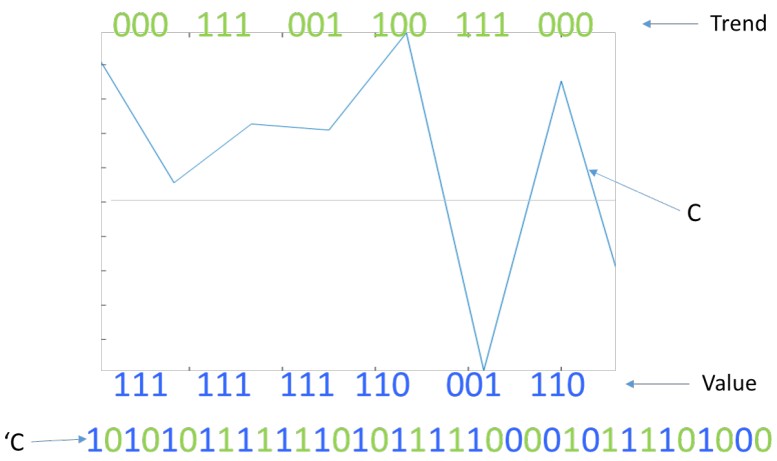

Figure 6.2: Proposed method illustration ($\omega = n$).

of the trend. In this particular scenario the result in some cases may be affected by some amount of noise due to the fact that we consider all the points when looking into the trend. The new representation is a string of size 2ω, where the odd positioned bits represent the symbol reflecting the value and the even positioned bits reflect the trend information. Having a dedicated position for the value and trend allow for more flexibility in the case one would want to manipulate or put more consideration into one set without impacting the other.

In a more general perspective Fig. 6.3 illustrates our proposed representation technique in the case where the number of segments ω is less than the length n of the original UTS. This scenario reflects the situation where the time series sequence is divided into segments of equal length. The middle point of each segment is chosen and the linear regression of the position of the curve corresponding to the midpoint is assessed and given a symbol based on whether it is above or below the mean. Consecutive midpoints are compared to get a sense of the trend and a symbol is allocated, to reflect the direction of the time series.

One of the advantages for our proposed representation technique is that it provides dedicated positions for the value and trend, and hence allows for the flexibility to allocate more weight into one set without affecting the other.

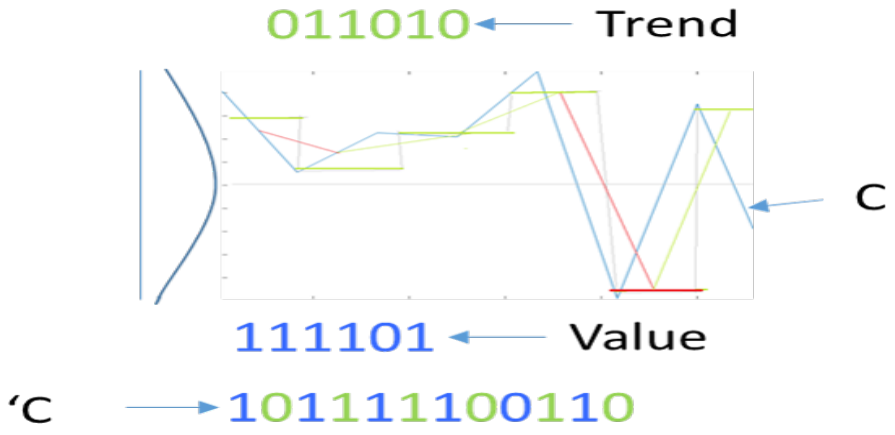

Figure 6.3: Proposed method illustration ($\omega << n$).

6.4.1 Similarity Measure

The second step in our proposed technique consists of measuring the similarity between time series.

Going through the symbolic transformation and computing the symbolic correlation first, will provide a better performance in speed and space, with the following advantages:

- no pairwise computation of the Pearson correlation on numerical values will be required among all pairs of series within the dataset
- Boolean operations are faster to perform (due to hardware support) and require less memory space during computation

Definition 6.1. *(Symbolic correlation) - Given binary symbolic series* $\hat{X}^i = <s^i_1, s^i_2, \ldots, s^i_{2n}>$ *and* $\hat{X}^j = <s^j_1, s^j_2, \ldots, s^j_{2n}>$*, and a correlation threshold* ε*, the symbolic correlation value* $\hat{\rho}$ *of the two series is:*

$$\hat{\rho} = 1 - (1/2n) * [\alpha \ (\sum_{1}^{n} s^i_{2k-1} \oplus s^j_{2k-1}) \ + \ \beta(\sum_{1}^{n} s^i_{2k} \oplus s^j_{2k})],$$

where $1 \leq k \leq n$*,* α *and* β *are the weights reflecting the importance of either group*

(trend or value). If $|\hat{\rho}| \geqslant \varepsilon$, the series are correlated within the threshold and included in the candidate set.

We are weighting dissimilarity between binary vectors, while allocating unequal importance to two different objects (bits where the "trends" match, and bits where the "values" match). We use and adapt the Hamming distance to a weighted dissimilarity measure for Binary variables to assess the correlation between the symbolic series.

The Hamming distance [31] applied to Boolean strings X and Y over $\{0,1\}^n$ can be seen as measuring the distance between Boolean vectors, expressed as follows:

$$d_{HD}(X,Y) = \sum_{1}^{n} X(i) \oplus Y(i)$$

where $\oplus$ is the boolean XOR operation applied to corresponding bit strings $X(i)$ and $Y(i)$ of the boolean vectors.

The Hamming distance returns the number of bit entries in which the boolean vectors differ.

We note that within each symbolic string, the even and odd bit positions are different in nature, representing respectively entries for values and trends. They can be considered separately. Each symbolic string has n even entries and n odd entries. Comparing the bit entries at odd positions consist of computing their dissimilarity using:

$$\sum_{1}^{n}(s^i_{2k-1} \oplus s^j_{2k-1}), \quad and\, 0 \leq \sum_{1}^{n}(s^i_{2k-1} \oplus s^j_{2k-1}) \leq n,$$

which is equivalent to:

$$0 \leq \frac{\sum_1^n (s^i_{2k-1} \oplus s^j_{2k-1})}{n} \leq 1.$$

Comparing the bit entries at even positions consist of computing their dissimilarity using:

$$\sum_{1}^{n}(s^i_{2k} \oplus s^j_{2k}), \quad and\, 0 \leq \sum_{1}^{n}(s^i_{2k} \oplus s^j_{2k}) \leq n,$$

which is equivalent to:

$$0 \leq \frac{\sum_1^n (s_{2k}^i \oplus s_{2k}^j)}{n} \leq 1.$$

Considering two entire time series of length n, we can evaluate their correlation as follows:

$$\hat{\rho} = 1 - (1/2n) * [\alpha \ (\sum_1^n s_{2k-1}^i \oplus s_{2k-1}^j) \ + \ \beta(\sum_1^n s_{2k}^i \oplus s_{2k}^j)].$$

where α is the weight associated with the odd bit positions (which corresponds to the values of the time series), and β is the weight associated with the even bit positions (which corresponds to the trend of the time series).

We want to find the correlation candidate set that will be comprised of all symbolic series whose computed symbolic correlation would return a value greater than the provided correlation threshold.

Once the candidate set is established, we compute the Pearson correlation between the pairs within the candidate set to obtain their coefficient of correlation.

Computing Pearson correlations between the time series within the reduced candidate set allow for a faster identification of the set that satisfies the Pearson correlation threshold in a smaller environment.

6.5 Applying CTVR to Multivariate Time Series

In this section we explore the feasibility of applying the proposed CTVR technique to multivariate time series (MTS). We are interested in the problem of similarity search in MTS defined as follows:

Definition 6.2. *(Multivariate time series similarity search)*
Let $D = \{T_{n,m}^1, T_{n,m}^2 ..., T_{n,m}^q\}$ be a set of MTS, each of which containing n instances and m variables; and ϵ be a user specified threshold value. A MTS similarity search retrieves all pairs of times series T^i and T^j in D such that their correlation distance does not exceed ϵ, for $1 \leq i, j \leq q$.

The first step in this application consist of reducing the dimensionality of the MTS. We use the representation technique M2U seen in section 5.3 to transform all MTS to UTS. We subsequently use the CTVR technique discussed in section 6.4 to transform the newly generated univariate series to symbolic series and find correlated pairs in a second and third steps. As discussed in section 5.1.3, a fair amount of preprocessing of the MTS data is required as the dimensionality reduction relies on PCA. Here as well, the number k_{max}, of relevant principal components to retain from each MTS is identified as illustrated in Algorithm 5.1 of section 5.3. Algorithm 6.2 summarizes the steps required in carrying out similarity search within a given set of MTS using ESTMSS (the efficient and scalable technique for MTS similarity search).

Algorithm 6.2 - Efficient And Scalable Technique for MTS Similarity Search (ESTMSS)

Input: $D' = \{A^1_{n,m}, A^2_{n,m} ..., A^q_{n,m}\}$ a set of normalized MTS, θ (cumulative variance explained), ϵ a user specified Pearson correlation threshold.
Output: A set C of all pairs (A^i, A^j) in D' whose correlation is not less than ϵ.
begin

1: Estimate k_{max} using Algorithm 5.1.
   $k \leftarrow k_{max}$
2: **for** $i \leftarrow 1$ *to* q **do**
3: Transform the MTS to a UTS using M2U
   $D^U \leftarrow M2U(A^i)$
4: **end for**
5: Using CTVR, transform the series in D^U to symbolic series and find correlated pairs
   $C \leftarrow CTVR(D^U)$

end

6.6 Performance Evaluation

To evaluate the effectiveness of our solution approach, we developed the code in Matlab and conducted numerous experiments using benchmark datasets. For this we use a configured PC with Intel Quad core i7 2.00 GHz CPU, 8 GB main memory, running Windows 7.

6.6.1 Datasets

The experiments were ran on benchmark datasets drawn from several widely used repositories [3, 78, 44] in the current literature. Experiments and results pertaining to two of the used datasets are reviewed in this section.

The financial Market indices [78] individually present a certain number of stocks for which a weighted average is computed (often based on the stock capitals) to reflect their overall performance in the market.

We selected the UTS representatives of the following five market indices to compare: The Dow Jones Industrial Average (combining 30 stocks representative of the American market), the NASDAQ-100 (tracking 100 largest non-financial companies in the National Association of Securities Dealers Automated Quotations market), the FTSE100 (combining 100 companies with the largest capitalization traded in the London market), the Deutscher Aktien indeX (DAX) (including 30 German companies traded in the Frankfurt market), and the S&P or the Standard & Poor's 500 (index based on the market capitalizations of 500 large companies having common stock listed on the NYSE or NASDAQ).

The time period used in this experiment included 11 months from May 19th 2010 to April 18th 2011.

The Activity Recognition from Single Chest-Mounted Accelerometer dataset (ARFSCMA) [3] is intended for motion patterns and activity recognition. It was gathered from 15 participants performing 7 activities while wearing accelerometer mounted on their chests, at a sampling frequency of 52 Hz. This dataset is often used in classification, clustering, and similarity search problems due to its complexity.

The Australian language sign dataset(AUSLAN) [45] was gathered through two gloves, with 22 sensors while native AUSLAN speakers signed. The dataset contains 95 signs having 27 examples each, hence a total of 2565 of signs gathered. This dataset is well used in similarity search problems due to its complexity.

The INRIA Holidays images dataset(INRIA HID) [39] is a collection of images that have served in testing the robustness to various transformations: rotations, viewpoint and illumination changes, blurring, etc. The dataset contains 500 high

resolution image groups representing a large variety of scene types to incorporate diversity in representation.

6.6.2 Evaluation and Results

We designed experiments to assess the performance of the proposed technique. In this section, we review a set of experiments through which we evaluated how varying lengths of the PAA segments impacted the performance and recall of our algorithm. We also review the effect of varying length of the PAA segments according to the time series length in different datasets. We finally assess recall and precision while comparing our results to those from the clipping technique [81] for UTS.

In the case of MTS, we looked into runtime and precision while comparing our results to those from primarily three techniques: the Correlation Based Dynamical Time Warping (CBDTW) [4], the Dynamical Time Warping, and an implementation of the standard PCA transformation followed by an exact match computing a pairwise comparison of all series.

Another set of experiments allowed to assessed the influence of the change in dimensionality on runtime while accuracy was being preserved; those experiments further allowed us to investigate the proposed technique's scalability on large MTS datasets.

We finally reviewed the effect of varying length of the PAA segments according to the time series length in different datasets.

6.6.2.1 Impact of the number of segments on precision

The precision of similarity is determined by varying length of the segments according to the length of the time series in the dataset. In our experiments, the best precision is observed when the number of segments w is closer to the length of the time series for the value based transformation. This can be explained by the fact that a more detailed capture of the characteristics of the original time series is achieved in that case. On the other hand, it requires slightly wider segments for the trend to avoid incorporating much noise and outliers. Fig. 6.4 shows the particular case of a dataset,

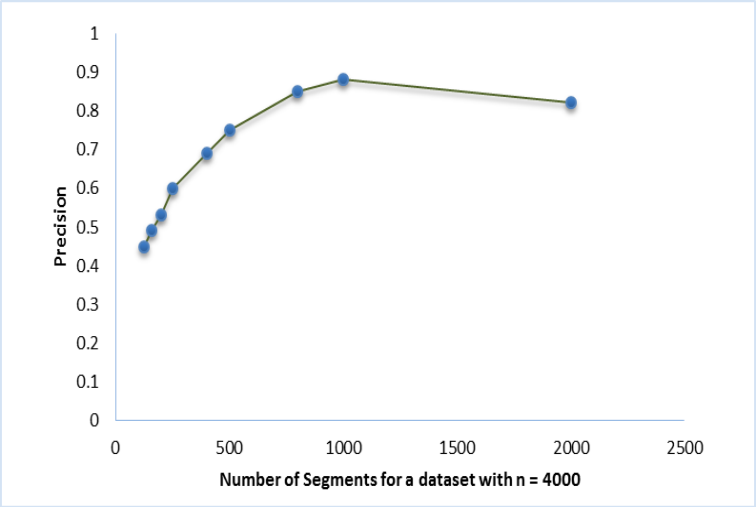

Figure 6.4: Precision as number of segments varies.

where the best precision was achieved for ω close to $n/4$, while the length of the time series is $n = 4000$. We investigated the impact that the trend or value had in uncovering correlations for this dataset. As illustrated in Fig. 6.5, a study of the relationship between the S&P 500 and NASDAQ-100 on the chosen time periods shows that for relatively smaller durations the value contributions was best to helping uncover correlations. However for longer durations the trend was best for uncovering correlations. Some datasets provide better results when the value is given more weight, while others rely more on the trend when it comes to uncovering similarities.

The results obtained from the indices dataset collected during May 19th 2010 to April 18th 2011 time period show a high correlation between S&P and Dow. This can be explained possibly by the fact that the S&P index is weighted, maintained and published by the same joint venture that published the Dow Jones Industrial average. NASDAQ although positively correlated to all indices, shows that association to a lesser degree during the chosen period of time. Results from our method compared to implementations of other methods show an improvement in run time and space while providing comparable precision and recall. For datasets where the trend is the most important factor, carefully picking the number of segments to best accommodate the trend help achieving better results, as it will avoid including noise and outliers.

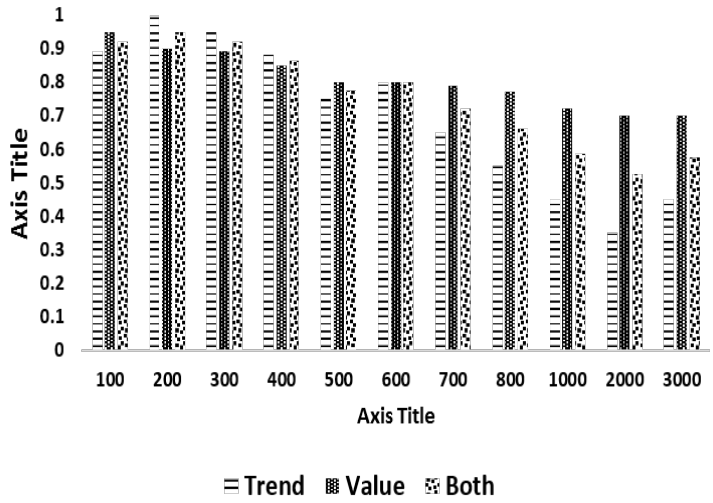

Figure 6.5: Identifying correlations while allocating more weight on value, trend or equally on both.

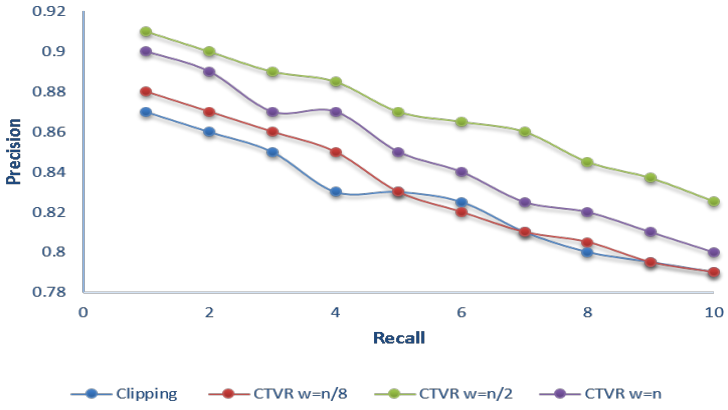

Figure 6.6: Precision/Recall on the ARFSCMA dataset for different techniques

6.6.2.2 Precision and Recall on the ARFSCMA dataset

The ARFSCMA dataset is known to be a complex dataset for similarity search problems due to the fact that different activities can share common components or patterns. Additional detailed information often helps differentiate those activities. In capturing the trend information in addition to the value information to be represented in symbolic time series, our proposed technique better captures the characteristics of the original time series, and increases the pruning power.

Experimental results summarizing an average for the seven activities as conducted by 15 participants are show on Fig. 6.6. We compare the precision/recall ratio achieved by the clipping technique [81] against that of our technique in 3 cases ($\omega = n/8$, $\omega = n/2$, $\omega = n$). The results for the case where $\omega = n/8$ are comparable to those for the clipping technique. As illustrated earlier, this can be explained by the fact that less details are captured from the original time series values in that case. The best results on this dataset were noted for the case $\omega = n/2$. An important advantage that our proposed technique provides when compared to the clipping technique, is the increased pruning power.

6.6.2.3 Execution time and precision as dimensionality increases on MTS

In this set of experiments we studied our performance on a large number benchmark datasets, and represent here results gathered from three of them.

The data from each market index is considered a MTS and represented as a matrix for processing using our technique. It is subsequently transformed into UTS for further processing using CTVR.

As the number of variables grows, so does the runtime. We can identify three phases within the runtime as illustrated in Fig. 6.7 for the stock indices dataset. The first phase allows to represent the MTS as a UTS, the second phase transforms the UTS to a symbolic representation, while the 3rd phase allows for the correlation computation.

Of the three phases, the transformation step seems to make up more of the run time. This is due to the fact that, in the first phase, we use a randomized PCA to

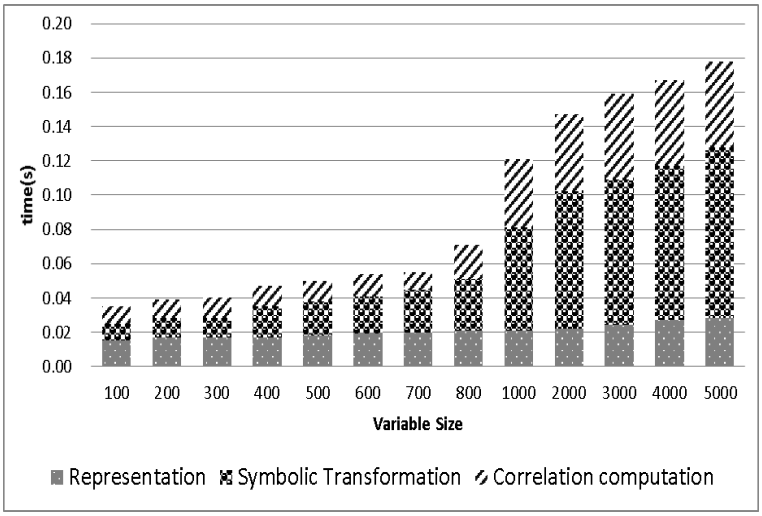

Figure 6.7: Run time for each step based on the number of variables

reduce the dimensionality, hence an enhanced execution speed is expected. And, in the third phase, we have a symbolic Boolean representation of the series. The correlation computation is based on bitwise operations, hence the speed of execution. We note an important improvement in computation time compared to peer techniques.

As the length of the time series sequence increases, we also note an increase in the precision. This can be explained by the fact that the symbolic representation is then more representative of the original time series for it captures more details.
Although a much smaller dataset, AUSLAN is known to be a complex dataset for similarity search problems due to the presence of a large number of zeroes making it harder to find correlations. However PCA is known to be very efficient in the processing of sparse matrices. For experiments conducted on this dataset, we selected the percentage of explained variance to be retained from each MTS to 90%. This ensures that much of the structure of the data is retained in the selected principal components. Given that this dataset has small length and a small number of variables, we selected the number of PAA segments to either n (the dimension of the time series) or n/2. For the correlation based dynamical time warping technique (CBDTW) [4], we set the number of segments to 20, for effective warping. Experimental results summarizing an average for over 100 category pairs of sign on the methods compared

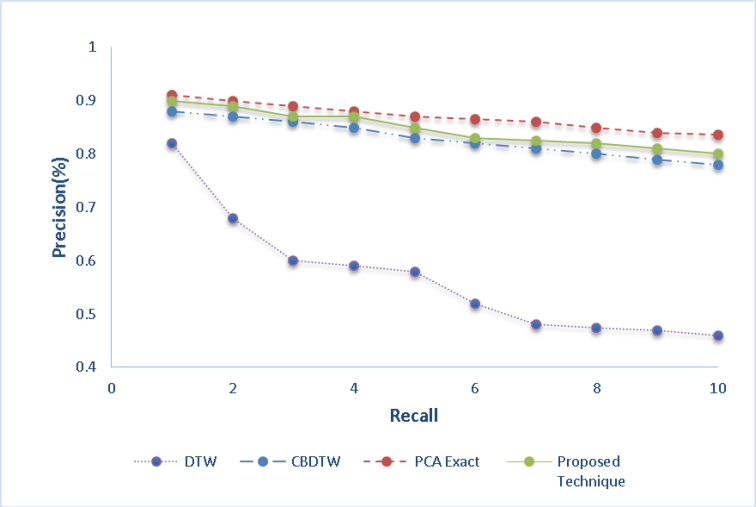

Figure 6.8: Precision/Recall on AUSLAN(MTS) for different algorithms

(DTW, CBDTW, PCA exact match, proposed technique) for this dataset show on Fig. 6.8, that our technique and the PCA exact match technique outperform the other two techniques in terms of precision and recall, while it substantially outperform the all techniques in terms of execution time.

Using the The INRIA Holidays images dataset [39], we concatenate images and look to study the effect of increasing the number of variables and the length of the time series on the proposed technique. Fig. 6.9 shows the accuracy graph, in function of the number of variables for the techniques that we compared. All three techniques provided good accuracy results on this dataset. Fig. 6.10 uses a logarithmic time scale and illustrates the time increase as dimensionality(dimensions and variables) increased.

In comparing our runtime against those of other techniques, our technique saves up to two orders of magnitude. For instance CBDTW required on average 40 times more than our technique on the same datasets and, was not able to process larger size matrices. While the PCA exact match runtime and our proposed technique appear to be close in runtime on Fig. 6.10, due to the relatively low number of dimensions/variables used in this test, in order to be able to include other peer techniques, the difference is expected to become much more important in larger dimension settings. The PCA

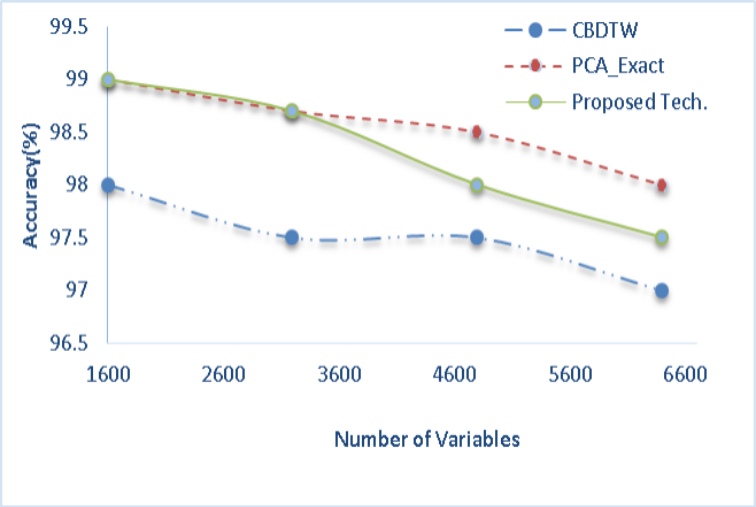

Figure 6.9: Accuracy on INRIA HID for different number of variables

exact match implementation is a version based on standard PCA and exact match of all pairs. It will present slightly better accuracy results than our proposed technique used with the Randomized PCA technique, it is however expected to be slower in large dimension settings.

6.6.2.4 The study of scalability on MTS

We have seen that our algorithm is comparable or out-performs peer techniques in terms of similarity search quality while presenting better running time for queries where the peer techniques can process the input size.

We further investigated the scalability of our approach to a framework where the dimensionality is much larger. More specifically how would the technique withstand a 1000% 10000% increase in dimensionality. Results gathered from the The INRIA Holidays images dataset are represented in this section. Fig. 6.11 shows that with a number of variables increased to 1000%, the running time increases by a factor of about 5.5. A time series length/dimension increase to 1000% leads to a the run time increase by a factor of 6.2. A 10000% increase in number of variables leads to an increase by a factor of 47 in time, while the same rate of increase in length/dimension leads to an increase by a factor of 41.6 in run time as seen in Fig. 6.12. We include

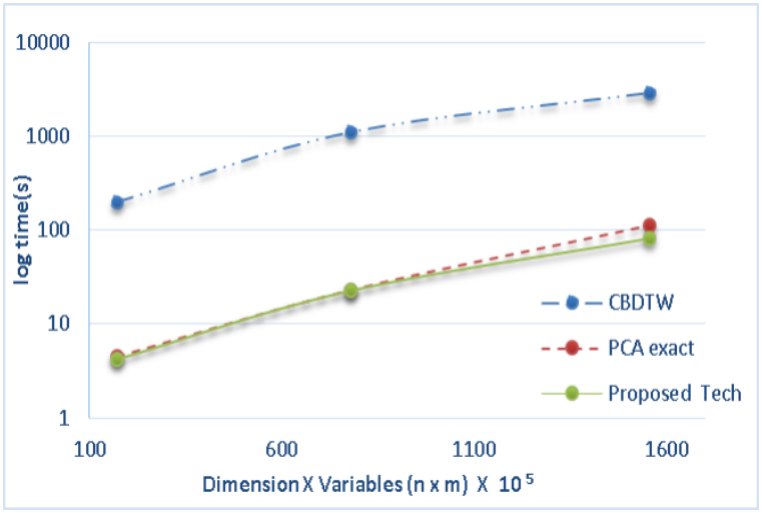

Figure 6.10: Runtime on INRIA HID as dimensionality increases

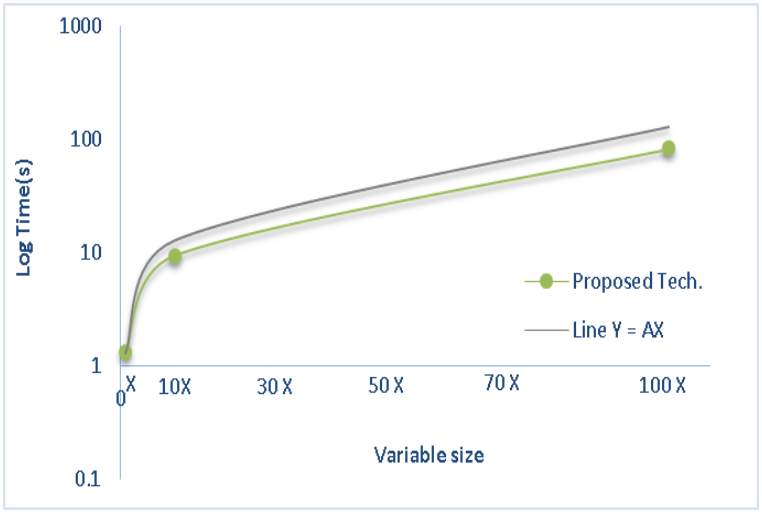

Figure 6.11: Run Time for larger number of variables

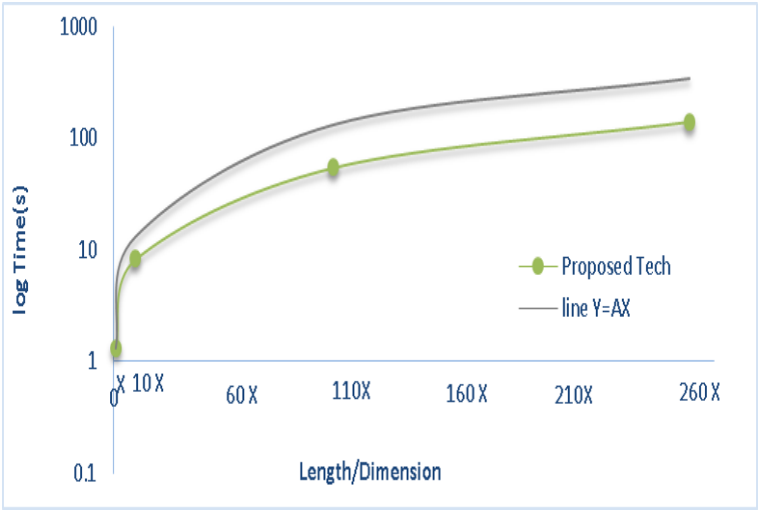

Figure 6.12: Run Time for longer time series

a line with linear behavior ($y = ax$) in Fig. 6.11 and Fig. 6.12 to further observe the limits of the running time. We can see that as we keep multiplying the dimension or variable by 10, the running time is linear and remains below the line y = ax. It appears to present an asymptotic behavior, although further empirical studies and theoretical analysis would be needed to obtain provable guarantees.

6.7 Summary

In this chapter we present an efficient technique for time series analysis and search. The proposed method allows for a reduction in memory requirement, an enhancement in performance, and an improvement in accuracy for similarity search in large data settings when compared to traditional techniques. In our future work, we intend to extend the proposed technique to streaming data and non linearly dependent data frameworks.

Chapter 7: Conclusion and Future Work

Research in multivariate time series (MTS) analysis and search has grown in importance, relevance and popularity in recent years. As data volume becomes prohibitively large, the traditional methods investigating meaningful patterns in such frameworks are no longer suitable for such massive high dimensional data (both in terms of size and number of variables). The objective of our work is to provide a framework where downstream pattern recognition tasks in general and similarity search operations in particular perform more efficiently in such settings.

In this dissertation, we presented a set of efficient and scalable unsupervised techniques for MTS analysis and search. The techniques primarily rely on the principal component analysis to learn and leverage the internal structure of the multivariate data for better results.

- We studied the problem of uncovering the most relevant and discriminative features in MTS and presented an unsupervised feature subset selection technique [48] based on statistics drawn from the Principal Component Analysis of the input data. While this technique leverages the desired properties of the PCA, it also retains the results interpretability by combining the advantages of feature extraction and selection techniques.

- In some frameworks, such as MTS data exhibiting sparse feature vectors, or MTS in some practical Bio-informatics applications where dependant features are known to work better in group than on their own feature subset selection

alone can come short in uncovering the most relevant and discriminative features in MTS. We developed an unsupervised feature subset selection and grouping technique [49], termed FRG (Feature Ranking and Grouping), that yields better results in such cases. The technique first relies on unsupervised learning through randomized Principal Component Analysis (PCA) to uncover influence and rank the features accordingly. Correlated features are then subsequently identified grouped and re-engineered into unique features to allow for a more efficient and scalable processing of the high dimensional MTS.

- We also developed a reduction and representation technique for MTS termed M2U [50](Multivariate to Univariate transformation). This technique is particularly important because, on one hand, the transformation takes into account the correlation between variables, while decreasing redundancy and noise, dimensionality and requirements in memory space and computation time. On the other hand substantial research and progress in making UTS pattern recognition tasks in general, and similarity search in particular, very efficient on large datasets has occurred in recent years [79, 88, 12, 71]. Our proposed representation will allow efficient UTS techniques to be easily extended to MTS.

- We developed a UTS transformation and representation technique, TVR [47](Trend and Value Representation) which extends the clipping technique [81] by incorporating the time series trend information, in addition to the value information to better capture the data characteristics and provide greater accuracy.

- We formulated of a weighted symbolic similarity measure measure based on a binary weighted dissimilarity measures [47] for mixed types of variables measuring different objects. Using this similarity measure along with TVR in the pruning phase allows to substantially reduce the search space in large dataset frameworks.

- We propose the use of the techniques M2U [50], TVR [47] and the proposed symbolic correlation measure based on a binary weighted dissimilarity measure

for mixed variables [47] in conjunction to devise ESTMSS, an efficient and scalable technique for multivariate time series similarity search.

Experiments were ran on benchmark data that has been extensively used in the literature; Our results show that the proposed technique outperformed peer techniques. The combination of these three techniques affords a suitable tool for uncovering similar multivariate time series in large data settings .

Our experimental evaluations of these techniques on a large number of application domains and extensively studied benchmark datasets indicate their performances in terms of accuracy speed and scalability when compared to state of art techniques. They provide simpler approaches to large MTS data processing, improvements in memory space requirements and computation time and improvements on precision and recall.

7.1 Future Work

While this thesis contributes to the field of MTS analysis and search in several aspects, the presented techniques can be extended to other types of data. We will look to generalize our techniques where possible. In addition, many opportunities for extending this work remain. This section presents some of the directions we intend to pursue in our future work.

Streaming multivariate time series: The techniques presented in this dissertation were developed for standard time series data at rest. We plan on extending this work to large streaming MTS, where the objective would be to devise some sort of models that adaptively uncovers information about the evolving internal structure of the data, for adaptive dimensionality reduction, similarity search, and improved prediction capabilities. Among the challenges that we foresee are: the known constraints that come in maintaining streams, the high dimensionality of the data, but also the need for a strategy that efficiently tracks and leverage correlations as maintaining pairwise correlations in such settings would not be efficient.

Uncertain multivariate time series: There is an increasing need for efficient and scalable MTS techniques that would process raw data as it is received, with all its imperfections (e.g. missing or erroneous data). In that framework, we particularly plan on extending our work to uncertain time series where the values of the series may be unavailable, imprecise or unknown at timestamps. We will first investigate feature selection and forecasting in this direction. The intention is to leverage feature selection in this case to improve prediction accuracy. We plan on using uncertain time series correlation analysis as an important step in our search for the most relevant features and expect the uncertain time series correlation proposed in [74] to be more suitable in this scenario. Indeed, standard time series techniques are often not well equipped for uncertain time series issues. The technique in [74] extends the sample Pearson correlation coefficient and uses the cumulative distribution function (CDF) of the random variables correlation rather than the exact correlation.

Non-Linear multivariate time series: Our proposed contributions are currently better suited for data that has Gaussian (normal) distribution, and with linear dependencies. We plan to extend our work to include Non-linear data, and data for which the dependence structure goes beyond linear correlation. We particularly intend to investigate and leverage manifold learning and nonlinear dimensionality reduction techniques. In the case of the similarity measure we believe that techniques such as the Bhattacharyya distance, Mutual Information (MI) and Copula probability distribution among others could be good techniques to further investigate.

Theoretical analysis aspects: We plan to investigate why our techniques outperform peer techniques from a theoretical analysis perspective. For instance, in the case of the Weighted Scores and FRG techniques, sampling columns from the MTS according to their Weighted Scores empirically proves to provide a better matrix approximation than the state of art techniques. Although the intuition behind the computation of the variable weights provides some insights into the reason for that noted performance, it would be interesting to obtain provable guarantees.

References

[1] Robert J Alcock, Yannis Manolopoulos, et al. Time-series similarity queries employing a feature-based approach. 43, 48

[2] Torben G Andersen, Tim Bollerslev, Francis X Diebold, and Clara Vega. Real-time price discovery in global stock, bond and foreign exchange markets. *Journal of International Economics*, 73(2):251–277, 2007. 3

[3] Arthur Asuncion and David Newman. Uci machine learning repository, 2007. 83, 103

[4] Zoltán Bankó and János Abonyi. Correlation based dynamic time warping of multivariate time series. *Expert Systems with Applications*, 39(17):12814–12823, 2012. 24, 25, 77, 84, 104, 108

[5] Ella Bingham and Heikki Mannila. Random projection in dimensionality reduction: applications to image and text data. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 245–250. ACM, 2001. 76

[6] Verónica Bolón-Canedo, Noelia Sánchez-Maroño, and Amparo Alonso-Betanzos. Recent advances and emerging challenges of feature selection in the context of big data. *Knowledge-Based Systems*, 86:33–45, 2015. 35, 56

[7] Howard D Bondell and Brian J Reich. Simultaneous regression shrinkage, variable selection, and supervised clustering of predictors with oscar. *Biometrics*, 64(1):115–123, 2008. 57, 58

[8] Boyan Bonev. *Feature selection based on information theory*. Universidad de Alicante, 2010. 35, 56

[9] Melinda Borello. *Standardization and Singular Value Decomposition in Canonical Correlation Analysis*. PhD thesis, Pitzer College, 2013. 13

[10] Christos Boutsidis, Michael W Mahoney, and Petros Drineas. Unsupervised feature selection for principal components analysis. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 61–69. ACM, 2008. 34, 56

[11] Gavin Brown, Adam Pocock, Ming-Jie Zhao, and Mikel Luján. Conditional likelihood maximisation: a unifying framework for information theoretic feature selection. *Journal of Machine Learning Research*, 13(Jan):27–66, 2012. 41, 64

[12] Alessandro Camerra, Themis Palpanas, Jin Shieh, and Eamonn Keogh. isax 2.0: Indexing and mining one billion time series. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 58–67. IEEE, 2010. 6, 18, 74, 91, 114

[13] Tony F Chan. Rank revealing qr factorizations. *Linear algebra and its applications*, 88:67–82, 1987. 34, 56

[14] Lei Chen, M Tamer Özsu, and Vincent Oria. Robust and fast similarity search for moving object trajectories. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 491–502. ACM, 2005. 23

[15] Geoff Davis, Stephane Mallat, and Marco Avellaneda. Adaptive greedy approximations. *Constructive approximation*, 13(1):57–98, 1997. 29, 35

[16] Nuno Sergio Dias, M Kamrunnahar, Paulo Mateus Mendes, SJ Schiff, and José Higino Correia. Feature selection on movement imagery discrimination and attention detection. *Medical & biological engineering & computing*, 48(4):331–341, 2010. 35, 57

[17] Petros Drineas, Ravi Kannan, and Michael W Mahoney. Fast monte carlo algorithms for matrices iii: Computing a compressed approximate matrix decomposition. *SIAM Journal on Computing*, 36(1):184–206, 2006. 14, 33

[18] Petros Drineas, Michael W Mahoney, and S Muthukrishnan. Relative-error cur matrix decompositions. *SIAM Journal on Matrix Analysis and Applications*, 30(2):844–881, 2008. 29, 36, 43, 44, 65, 66

[19] Dryad. http://www.http://datadryad.org/. 39, 62

[20] Richard O Duda, Peter E Hart, and David G Stork. *Pattern classification*. John Wiley & Sons, 2012. 34, 43, 44, 65, 66

[21] Philippe Esling and Carlos Agon. Time-series data mining. *ACM Computing Surveys (CSUR)*, 45(1):12, 2012. 15

[22] Bilal Esmael, Arghad Arnaout, Rudolf K Fruhwirth, and Gerhard Thonhauser. Multivariate time series classification by combining trend-based and value-based approximations. In *Computational Science and Its Applications–ICCSA 2012*, pages 392–403. Springer, 2012. 19, 25, 73, 74

[23] JB Fourier. Théorie analytique de la chaleur, english translation by a, 1822. 10

[24] Dmitriy Fradkin and David Madigan. Experiments with random projections for machine learning. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 517–522. ACM, 2003. 14, 76

[25] Jerome Friedman, Trevor Hastie, Holger Höfling, Robert Tibshirani, et al. Pathwise coordinate optimization. *The Annals of Applied Statistics*, 1(2):302–332, 2007. 57

[26] Nir Friedman, Dan Geiger, and Moises Goldszmidt. Bayesian network classifiers. *Machine learning*, 29(2-3):131–163, 1997. 41, 64

[27] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *The Journal of Machine Learning Research*, 3:1157–1182, 2003. 29, 35, 41, 42, 55, 64, 67

[28] Alfred Haar. Zur theorie der orthogonalen funktionensysteme. *Mathematische Annalen*, 69(3):331–371, 1910. 11

[29] Nathan Halko, Per-Gunnar Martinsson, Yoel Shkolnisky, and Mark Tygert. An algorithm for the principal component analysis of large data sets. *SIAM Journal on Scientific computing*, 33(5):2580–2594, 2011. 33

[30] Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011. 14, 33, 56, 62, 75

[31] Richard W Hamming. Error detecting and error correcting codes. *Bell System technical journal*, 29(2):147–160, 1950. 100

[32] Min Han and Xiaoxin Liu. Feature selection techniques with class separability for multivariate time series. *Neurocomputing*, 110:29–34, 2013. 35, 57

[33] Lars Kai Hansen. Blind separation of noisy image mixtures. 2000. 13

[34] James V Haxby. Multivariate pattern analysis of fmri: the early beginnings. *Neuroimage*, 62(2):852–855, 2012. 3

[35] Bing Hu. Mining time series data: Moving from toy problems to realistic deployments. 2013. 18

[36] Bing Hu, Yanping Chen, Jamaluddin Zakaria, Liudmila Ulanova, and Eamonn Keogh. Classification of multi-dimensional streaming time series by weighting each classifier's track record. In *Data Mining (ICDM), 2013 IEEE 13th International Conference on*, pages 281–290. IEEE, 2013. 18, 21

[37] Gordon P Hughes. On the mean accuracy of statistical pattern recognizers. *Information Theory, IEEE Transactions on*, 14(1):55–63, 1968. 35, 56

[38] Aapo Hyvärinen and Erkki Oja. Independent component analysis: algorithms and applications. *Neural networks*, 13(4):411–430, 2000. 76

[39] H Jegou, M Douze, and C Schmid. Inria holidays dataset, 2008. 83, 84, 103, 109

[40] William B Johnson and Joram Lindenstrauss. Extensions of lipschitz mappings into a hilbert space. *Contemporary mathematics*, 26(189-206):1, 1984. 13, 76

[41] Ian Jolliffe. *Principal component analysis*. Wiley Online Library. 29, 36, 75

[42] Ian T Jolliffe. Discarding variables in a principal component analysis. i: Artificial data. *Applied statistics*, pages 160–173, 1972. 35, 36, 43, 44, 45, 65, 66, 69

[43] Sungkyu Jung, J Stephen Marron, et al. Pca consistency in high dimension, low sample size context. *The Annals of Statistics*, 37(6B):4104–4130, 2009. 31

[44] M Just and T Mitchell. Starplus fmri data. *URL http://www. cs. cmu. edu/afs/cs. cmu. edu/project/theo-81/www*, 2001. 39, 62, 103

[45] Mohammed Waleed Kadous. *Temporal classification: Extending the classification paradigm to multivariate time series*. PhD thesis, The University of New South Wales, 2002. 83, 103

[46] Tamer Kahveci, Ambuj Singh, and Aliekber Gurel. Similarity searching for multi-attribute sequences. In *Scientific and Statistical Database Management, 2002. Proceedings. 14th International Conference on*, pages 175–184. IEEE, 2002. 19, 73

[47] Aminata Kane. Trend and value based time series representation for similarity search. In *Multimedia Big Data (BigMM), 2017 IEEE Third International Conference on*, pages 252–259. IEEE, 2017. 6, 7, 114, 115

[48] Aminata Kane and Nematollaah Shiri. Selecting the top-k discriminative features using principal component analysis. In *Data Mining Workshops*

(ICDMW), 2016 IEEE 16th International Conference on, pages 639–646. IEEE, 2016. 3, 5, 13, 60, 65, 78, 113

[49] Aminata Kane and Nematollaah Shiri. Feature selection and grouping for multivariate time series using pca. In *Proceedings of the KDD Workshop on Data Driven Discovery*, pages xx–xx. ACM, 2017. 5, 114

[50] Aminata Kane and Nematollaah Shiri. Multivariate time series representation and similarity search using pca. In *Industrial Conference on Data Mining*, pages 122–136. Springer, 2017. 6, 7, 114

[51] Leonidas Karamitopoulos, Georgios Evangelidis, and Dimitris Dervos. Pca-based time series similarity search. In *Data Mining*, pages 255–276. Springer, 2010. 3, 24, 77

[52] Kendrick N Kay, Thomas Naselaris, Ryan J Prenger, and Jack L Gallant. Identifying natural images from human brain activity. *Nature*, 452(7185):352, 2008. 3

[53] Benjamin Kedem. Estimation of the parameters in stationary autoregressive processes after hard limiting. *Journal of the American Statistical Association*, 75(369):146–153, 1980. 6, 16, 93

[54] Benjamin Kedem and Eric Slud. On goodness of fit of time series models: An application of higher order crossings. *Biometrika*, 68(2):551–556, 1981. 6, 16, 93

[55] E Keogh. Ucr time series archive www. cs. ucr. edu/˜ eamonn, 2006. 39, 42, 62, 83

[56] E Keogh. Machine learning in time series databases (and everything is a time series!). In *Tutorial at the AAAI Int. Conf. on Artificial Intelligence*, 2011. 2

[57] Eamonn Keogh, Kaushik Chakrabarti, Michael Pazzani, and Sharad Mehrotra. Dimensionality reduction for fast similarity search in large time series databases. *Knowledge and information Systems*, 3(3):263–286, 2001. 24, 94

[58] Eamonn Keogh, Li Wei, Xiaopeng Xi, Michail Vlachos, Sang-Hee Lee, and Pavlos Protopapas. Supporting exact indexing of arbitrarily rotated shapes and periodic time series under euclidean and warping distance measures. *The VLDB JournalThe International Journal on Very Large Data Bases*, 18(3):611–630, 2009. 2

[59] Seyoung Kim and Eric P Xing. Statistical estimation of correlated genome associations to a quantitative trait network. *PLoS Genet*, 5(8):e1000587, 2009. 57

[60] Ron Kohavi and George H John. Wrappers for feature subset selection. *Artificial intelligence*, 97(1):273–324, 1997. 41

[61] Igor Kononenko, Edvard Simec, and Marko Robnik-Sikonja. Overcoming the myopia of inductive learning algorithms with relieff. *Applied Intelligence*, 7(1):39–55, 1997. 35, 43, 65

[62] Catherine Krier, Damien Francois, Fabrice Rossi, and Michel Verleysen. Feature clustering and mutual information for the selection of variables in spectral data. In *ESANN*, pages 157–162, 2007. 58

[63] Alexander V Lebedev. Mlsp 2014 schizophrenia classification challenge. 2014. 39, 42, 49, 62

[64] P Legendre and L Legendre. Numerical ecology (second english edition) elsevier science bv, 1998. 60

[65] Caiyan Li and Hongzhe Li. Network-constrained regularization and variable selection for analysis of genomic data. *Bioinformatics*, 24(9):1175–1182, 2008. 57

[66] M. Lichman. UCI machine learning repository, 2013. 39, 41, 42, 43, 44, 45, 62, 67, 69

[67] Jessica Lin, Eamonn Keogh, Stefano Lonardi, and Bill Chiu. A symbolic representation of time series, with implications for streaming algorithms. In *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, pages 2–11. ACM, 2003. 24, 26, 93

[68] Mina Maleki and Luis Rueda. Classification via correlation-based feature grouping. In *Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), 2015 IEEE Conference on*, pages 1–6. IEEE, 2015. 58

[69] Simon Malinowski, Thomas Guyet, René Quiniou, and Romain Tavenard. 1d-sax: A novel symbolic representation for time series. In *Advances in Intelligent Data Analysis XII*, pages 273–284. Springer, 2013. 94

[70] Vasileios Megalooikonomou, Qiang Wang, Guo Li, and Christos Faloutsos. A multiresolution symbolic representation of time series. In *Data Engineering, 2005. ICDE 2005. Proceedings. 21st International Conference on*, pages 668–679. IEEE, 2005. 93

[71] Abdullah Mueen, Suman Nath, and Jie Liu. Fast approximate correlation for massive time-series data. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pages 171–182. ACM, 2010. 6, 74, 114

[72] Balas Kausik Natarajan. Sparse approximate solutions to linear systems. *SIAM journal on computing*, 24(2):227–234, 1995. 29, 35

[73] Satoshi Niijima and Yasushi Okuno. Laplacian linear discriminant analysis approach to unsupervised feature selection. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 6(4):605–614, 2009. 36

[74] Mahsa Orang and Nematollaah Shiri. Correlation analysis techniques for uncertain time series. *Knowledge and Information Systems*, 50(1):79–116, 2017. 116

[75] Spiros Papadimitriou, Jimeng Sun, and Christos Faloutsos. Streaming pattern discovery in multiple time-series. In *Proceedings of the 31st international conference on Very large data bases*, pages 697–708. VLDB Endowment, 2005. 19

[76] Karl Pearson. Mathematical contributions to the theory of evolution. xix. second supplement to a memoir on skew variation. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, pages 429–457, 1916. 61, 75, 82, 92

[77] Hanchuan Peng, Fuhui Long, and Chris Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence*, 27(8):1226–1238, 2005. 43, 44, 65, 66

[78] Quandl. http://www.quandl.com/help/api. 39, 64, 103

[79] Thanawin Rakthanmanon, Bilson Campana, Abdullah Mueen, Gustavo Batista, Brandon Westover, Qiang Zhu, Jesin Zakaria, and Eamonn Keogh. Searching and mining trillions of time series subsequences under dynamic time warping. pages 262–270, 2012. 6, 16, 18, 19, 74, 91, 114

[80] Michalis Raptis, Darko Kirovski, and Hugues Hoppe. Real-time classification of dance gestures from skeleton animation. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics symposium on computer animation*, pages 147–156. ACM, 2011. 21, 22

[81] Chotirat Ratanamahatana, Eamonn Keogh, Anthony J Bagnall, and Stefano Lonardi. A novel bit level time series representation with implication of similarity search and clustering. In *Advances in knowledge discovery and data mining*, pages 771–777. Springer, 2005. xii, 6, 14, 15, 16, 20, 93, 94, 95, 104, 107, 114

[82] Dalton Rosario. A semiparametric model for hyperspectral anomaly detection. *Journal of Electrical and Computer Engineering*, 2012:6, 2012. 19, 25, 73

[83] Davide Roverso. Plant diagnostics by transient classification: The aladdin approach. *International Journal of Intelligent Systems*, 17(8):767–790, 2002. 84

[84] Mehmet Sayal. Detecting time correlations in time-series data streams. 2004. 20

[85] Henry Schütze, Thomas Martinetz, Silke Anders, and Amir Madany Mamlouk. A multivariate approach to estimate complexity of fmri time series. In *International Conference on Artificial Neural Networks*, pages 540–547. Springer, 2012. 3

[86] Reza Sherkat and Davood Rafiei. On efficiently searching trajectories and archival data for historical similarities. *Proceedings of the VLDB Endowment*, 1(1):896–908, 2008. 19, 25, 73

[87] Jin Shieh and Eamonn Keogh. i sax: indexing and mining terabyte sized time series. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 623–631. ACM, 2008. 18

[88] Jin Shieh and Eamonn Keogh. isax: disk-aware mining and indexing of massive time series datasets. *Data Mining and Knowledge Discovery*, 19(1):24–57, 2009. 6, 91, 114

[89] Nguyen Thanh Son and Duong Tuan Anh. Time series similarity search based on middle points and clipping. In *Data Mining and Optimization (DMO), 2011 3rd Conference on*, pages 13–19. IEEE, 2011. 93

[90] Fengxi Song, Zhongwei Guo, and Dayong Mei. Feature selection using principal component analysis. In *System Science, Engineering Design and Manufacturing Informatization (ICSEM), 2010 International Conference on*, volume 1, pages 27–30. IEEE, 2010. 29, 43, 44, 45, 65, 66, 69

[91] Jingping Song, Zhiliang Zhu, and Chris Price. Feature grouping for intrusion detection system based on hierarchical clustering. In *International Conference on Availability, Reliability, and Security*, pages 270–280. Springer, 2014. 58

[92] Stephan Spiegel, Julia Gaebler, Andreas Lommatzsch, Ernesto De Luca, and Sahin Albayrak. Pattern recognition and classification for multivariate time series. In *Proceedings of the fifth international workshop on knowledge discovery from sensor data*, pages 34–42. ACM, 2011. 18, 26

[93] Laerd Statistics. How to perform a principal components analysis (pca) in spss statistics. https://statistics.laerd.com/spss-tutorials/principal-components-analysis-pca-using-spss-statistics.php. 30

[94] Yoshiki Tanaka, Kazuhisa Iwamoto, and Kuniaki Uehara. Discovery of time-series motif from multi-dimensional data based on mdl principle. *Machine Learning*, 58(2-3):269–300, 2005. 24, 76, 77

[95] Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):91–108, 2005. 57

[96] Allan Tucker, Stephen Swift, and Xiaohui Liu. Variable grouping in multivariate time series via correlation. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 31(2):235–245, 2001. 57, 58

[97] Edward R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, CT, USA, 1986. 2

[98] Michail Vlachos, George Kollios, and Dimitrios Gunopulos. Discovering similar multidimensional trajectories. In *Data Engineering, 2002. Proceedings. 18th International Conference on*, pages 673–684. IEEE, 2002. 22

[99] Qiang Wang, Vasileios Megalooikonomou, and Guo Li. A symbolic representation of time series. In *ISSPA*, pages 655–658, 2005. 20

[100] Yongli Wang, Gongxuan Zhang, and Jiang-Bo Qian. Approxcca: An approximate correlation analysis algorithm for multidimensional data streams. *Knowledge-Based Systems*, 24(7):952–962, 2011. 3, 21

[101] Lior Wolf and Amnon Shashua. Feature selection for unsupervised and supervised inference: The emergence of sparsity in a weight-based approach. *Journal of Machine Learning Research*, 6(Nov):1855–1887, 2005. 35

[102] YahooFinance. http://finance.yahoo.com/. 42

[103] Kiyoung Yang and Cyrus Shahabi. A pca-based similarity measure for multivariate time series. In *Proceedings of the 2nd ACM international workshop on Multimedia databases*, pages 65–74. ACM, 2004. 3, 19, 23, 73, 77

[104] Kiyoung Yang and Cyrus Shahabi. On the stationarity of multivariate time series for correlation-based data analysis. In *Data Mining, Fifth IEEE International Conference on*, pages 4–pp. IEEE, 2005. 24

[105] Kiyoung Yang, Hyunjin Yoon, and Cyrus Shahabi. Cle ver: A feature subset selection technique for multivariate time series. In *Advances in Knowledge Discovery and Data Mining*, pages 516–522. Springer, 2005. 43, 44, 65, 66

[106] Kiyoung Yang, Hyunjin Yoon, and Cyrus Shahabi. A supervised feature subset selection technique for multivariate time series. 2005. 35, 57, 84

[107] Byoung-Kee Yi and Christos Faloutsos. Fast time sequence indexing for arbitrary lp norms. VLDB, 2000. 24, 94

[108] Hyunjin Yoon, Kiyoung Yang, and Cyrus Shahabi. Feature subset selection and feature ranking for multivariate time series. *IEEE transactions on knowledge and data engineering*, 17(9):1186–1198, 2005. 35, 57

[109] Seung-Chul Yoon and Bosoon Park. Hyperspectral image processing methods. In *Hyperspectral Imaging Technology in Food and Agriculture*, pages 81–101. Springer, 2015. 29, 36

[110] Yiteng Zhai, Yew-Soon Ong, and Ivor W Tsang. Making trillion correlations feasible in feature grouping and selection. *IEEE transactions on pattern analysis and machine intelligence*, 38(12):2472–2486, 2016. 6, 55, 57, 58

[111] Tiancheng Zhang, Dejun Yue, Yu Gu, Yi Wang, and Ge Yu. Adaptive correlation analysis in stream time series with sliding windows. *Computers & Mathematics with Applications*, 57(6):937–948, 2009. 20, 93, 94

[112] Guodong Zhao and Sanming Liu. Estimation of discriminative feature subset using community modularity. *Scientific reports*, 6, 2016. 44, 69

[113] Leon Wenliang Zhong and James T Kwok. Efficient sparse modeling with automatic feature grouping. *IEEE transactions on neural networks and learning systems*, 23(9):1436–1447, 2012. 57

[114] Yunyue Zhu. *High performance data mining in time series: techniques and case studies*. PhD thesis, New York University, 2004. 13, 14, 61, 76, 82, 92

[115] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005. 57